Planning with Consistency Models for Model-Based Offline Reinforcement Learning

Guanquan Wang
guanquan-wang@g.ecc.u-tokyo.ac.jp
Department of Information and Communication Engineering
The University of Tokyo

Takuya Hiraoka
takuya-h1@nec.com
NEC Corporation, Tokyo, Japan

Yoshimasa Tsuruoka
yoshimasa-tsuruoka@g.ecc.u-tokyo.ac.jp
Department of Information and Communication Engineering
The University of Tokyo

Abstract

This paper introduces consistency models in the problem of sequential decision-making. Previous work applying diffusion models to planning within a model-based reinforcement learning framework often struggles with high computational cost during the inference process, due to its reliance on iterative reverse diffusion processes. Consistency models, known for their computational efficiency, have already shown promise in reinforcement learning within the actor-critic algorithm. Therefore, we combine guided consistency distillation with a continuous-time diffusion model in the framework of Decision Diffuser. Our approach, named Consistency Planning, combines the robust planning capabilities of diffusion models with the speed of consistency models. We validate our method on gym tasks in the D4RL framework, demonstrating that compared with its diffusion model counterparts, our method achieves more than 12-fold increase in speed without any loss in performance.

1 Introduction

In recent years, significant strides have been made in high-resolution image generation through the advancement of diffusion-based generative models. Similarly, in offline reinforcement learning (RL) settings, deriving effective policies from pre-existing offline datasets can be simplified to the task of developing a probabilistic model for trajectory prediction, an area where diffusion-based generative models have proven to be highly successful. Existing models such as Diffuser (Janner et al., 2022) and Decision Diffuser (Ajay et al., 2022) underscore the efficacy of applying diffusion models to planning within model-based RL frameworks. In Diffuser, a diffusion model is trained on the trajectories in offline datasets, and then a separate classifier model is trained to predict the cumulative rewards of trajectory samples. During the inference process, the diffusion model with classifier guidance is employed to sample trajectories with high returns. Likewise, Decision Diffuser introduces a conditional diffusion model with state sequences as input, utilizing the return as a conditioning variable for classifier-free guidance during sampling. Moreover, incorporating only the state sequence—excluding the action sequence—Decision Diffuser trains an extra inverse dynamic model to infer actions.

Parallel to these developments, diffusion models have been adapted to model-free reinforcement learning scenarios, as illustrated by Diffusion-QL (Wang et al., 2022) and further enhanced in the efficient diffusion policy (EDP) (Kang et al., 2024). Diffusion-QL, utilizing a denoising diffusion probabilistic model (DDPM) (Ho et al., 2020), frames the diffusion model as a policy representation, conditioned on states with actions as outputs. It integrates Q-learning guidance into the reverse diffusion process to seek optimal actions. Despite its advancements, Diffusion-QL faces limitations in computational efficiency and its exclusive application within TD3-type algorithms (Fujimoto & Gu, 2021). EDP addresses these issues by introducing an action approximation trick during training, applying the DPM-solver, and approximating policy likelihood via the evidence lower bound in DDPM to overcome the limitations of Diffusion-QL.

Although the integration of diffusion models within both model-based and model-free frameworks in the offline RL setting has been extensively explored and enhanced, a significant challenge remains in their application, particularly in real-time decision-making contexts. This challenge stems from the diffusion models' reliance on iterative sampling processes, which can be computationally intensive and slow, thus restricting their use in scenarios that require rapid inference. For instance, in robot arm control (Chi et al., 2023), standard diffusion-based control can only make decisions at around 10Hz. However, this is insufficient for tasks requiring agile motion planning at 20Hz (Smith et al., 2023), 30Hz (Peng et al., 2020), or even higher.

Recent endeavors by Song et al. (2023) have introduced consistency models, a novel class of generative models that significantly enhance computational efficiency without sacrificing the expressiveness and flexibility that make diffusion models appealing for reinforcement learning.

In the model-free RL domain, consistency models have demonstrated promising results as policy representations, particularly in offline and offline-to-online RL settings (Ding & Jin, 2023). These developments underscore the consistency models' capability to effectively navigate the challenges of learning from fixed datasets, indicating their potential to achieve performance comparable to diffusion-based approaches, with higher computational efficiency.

However, in offline RL settings, model-free methods using Q-network face the challenges due to overestimated Q-values for out-of-distribution actions (Kumar et al., 2020; Levine et al., 2020). In the context of online RL, the problem is self-corrected as the policy interacts with the environment; an action perceived as favorable might receive a low reward, thus adjusting the policy. However, in offline RL, such corrections are not readily achievable, leading the learned Q-function to often guide the diffusion model towards potentially sub-optimal actions. Therefore, given the computational efficiency of consistency models and the proven effectiveness of diffusion models in trajectory prediction, this paper aims to explore how consistency models can augment model-based RL with classifier-free guidance in offline setting, bypassing the necessity of learning a Q-function by conditioning the consistency models on returns.

The goal of this paper is to bridge this gap by proposing a novel approach that merges the computational efficiency of consistency models with the planning capabilities inherent in Decision Diffuser. By integrating consistency models into the trajectory optimization process, we aim to leverage their computational advantages to enhance the speed of planning. Our experiments, conducted in offline RL settings, embed a conditional consistency model in the Decision Diffuser algorithm, evaluating with consistency distillation methods. Specifically, the Consistency Model employs guided consistency distillation from a score-based diffusion model (Karras et al., 2022; Ho & Salimans, 2022; Luo et al., 2023) pretrained on offline trajectories datasets.

In summary, our contribution is proposing Consistency Planning, a novel offline RL algorithm that extends the applicability of consistency models to model-based RL. We evaluate Consistency Planning on D4RL benchmark tasks (Fu et al., 2020) for offline RL, demonstrating that this method can achieve performance comparable to its diffusion model counterparts across the majority of tasks, with a notably faster sampling process.

2 Related Work

2.1 Diffusion Models

Diffusion models have emerged as a powerful approach for generating high-quality image and text data, as demonstrated by previous studies (Saharia et al., 2022; Nichol & Dhariwal, 2021). The data sampling process is formulated as an iterative denoising procedure, introduced by Sohl-Dickstein et al. (2015) and further developed by Ho et al. (2020). Parameterizing the gradients of the data distribution serves as an alternative interpretation of this denoising procedure, aiming to optimize the score matching objective, as elucidated by Hyvärinen & Dayan (2005). This positions the approach within the domain of Energy-Based Models, as evidenced by the contributions of Du & Mordatch (2019), Nijkamp et al. (2019), and Grathwohl et al. (2020).

Prior work (Nichol & Dhariwal, 2021) has implemented a classifier to enable the generation of images based on conditional information (e.g., text), which is called classifier guidance. However, more recent studies (Ho & Salimans, 2022), propose classifier-free guidance, which relies on the gradients from an implicit classifier, derived from the score function differences between conditional and unconditional models. This approach has proven to enhance the quality of conditional samples over classifier guidance methods. These advancements predominantly focus on text and image generation.

2.2 Diffusion Models in Reinforcement Learning

Diffusion models offer a versatile approach for data augmentation in reinforcement learning. SynthER (Lu et al., 2024) employs unguided diffusion models to enhance both offline and online RL datasets, subsequently utilized by model-free off-policy algorithms. Although this approach boosts performance, SynthER's reliance on unguided diffusion to approximate the behavior distribution faces challenges due to distributional shift. Similarly, MTDiff (He et al., 2024) implements unguided data generation in multitask environments.

Additionally, diffusion models have been adapted for training world models. For instance, Alonso et al. (2023) use diffusion to train world models, achieving precise predictions of future observations. However, this method does not model entire trajectories, leading to compounded errors and lack of policy guidance. In a related effort, Rigter et al. (2023) integrate policy guidance to enhance a diffusion world model in online RL. Jackson et al. (2024) concentrate on offline RL, providing a theoretical framework and rationale for the trajectory distribution shaped by policy guidance.

Diffusion models (Ho et al., 2020; Song et al., 2020) have also been adapted for policy representation in RL, capturing the multi-modal distributions in offline datasets. Specifically, Diffusion-QL (Wang et al., 2022), applies the diffusion model within the framework of both Q-learning and Behavior Cloning (BC) for policy representation. However, the main limitation of Diffusion-QL is that it demonstrates computational inefficiency due to the necessity of processing both forward and backward through the entire Markov chain during training. To alleviate these issues, Kang et al. (2024) introduces action approximation, eliminating the need to execute the denoising process during the training process.

Diffusion models have been employed in recent studies for human behavior imitation learning (Pearce et al., 2023) and trajectory generation in offline RL. Trajectories that include states and actions are generated by Diffuser (Janner et al., 2022), using an unconditional diffusion model, guided by a reward function trained on noisy state-action pairs. Decision Diffuser (Ajay et al., 2022) models the trajectories with the dataset using a unified, conditional generative model, avoiding separate training a classifier for reward functions.

3 Preliminary

3.1 Reinforcement Learning Problem Setting

The sequential decision-making problem is defined as a Markov decision process (MDP): $M = \{S, A, P, R, \gamma, d_0\}$, where S and A are the state space and the action space respectively, $P : S \times A \to S$ represents the transition function, $R : S \times A \times S \to \mathbb{R}$ denotes the reward function, $\gamma \in [0, 1)$ is the discount factor, d_0 is the initial state distribution. The goal of RL is to learn policy $\pi_\theta(a|s)$ to maximize the expected sum of discounted rewards $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$.

3.2 Consistency Models

Diffusion models operate by introducing Gaussian perturbations to transform data into noise, followed by generating data samples through a series of sequential denoising steps. Song et al. (2020) introduce a stochastic differential equation (SDE) framework that ensures the maintenance of the desired distribution as sample x evolves over time. The consistency models proposed by Song et al. (2023) recover the original data sample by solving a corresponding probability flow ordinary differential equation (ODE): $\frac{\mathrm{d}x_t}{\mathrm{d}t} = -t\nabla \log_t p_t(x)$, where $p_t(x) = p_{data}(x) \otimes \mathcal{N}(0, t^2\mathbf{I})$, $p_{data}(x)$ represents the original data distribution, $t \in [0, T]$ is the time period. The data generation process in this framework reverses along the trajectory $\{\hat{x}_t\}_{t\in[\epsilon,T]}$ of the ODE, starting from random initial samples $\hat{x}_T \sim \mathcal{N}\left(0, T^2\mathbf{I}\right)$ where ϵ is a minimal constant close to 0 to address numerical issues at the boundary.

To accelerate the sampling process in diffusion models, the consistency model significantly reduces the number of steps required for sampling compared to the original diffusion model, without substantially compromising the model's performance. This is achieved by approximating a parameterized consistency function, $f_\theta : (x_t, t) \to x_\epsilon$, which maps a noisy sample x_t at step t back to the original sample x_ϵ.

This approach differs from the diffusion model, which utilizes a step-by-step denoising function $p_\theta\left(x_{t-1} \mid x_t\right)$, for the reverse diffusion process. Slightly different from the original consistency model, this paper focuses on a conditional distribution, so the consistency function is modified to $f_\theta(x_t, t, c)$, where c denotes the condition variable.

4 Planning with Consistency Model

Diffusion Model Training. This paper explores the integration of consistency models, which are trained by distillation from a pre-trained diffusion model, into the planning architecture of Decision Diffuser. The original consistency models draw from the principles of score-based diffusion models (Song et al., 2020; Karras et al., 2022), making direct distillation from the discrete-time model used in Decision Diffuser ineffective.

As outlined by Ajay et al. (2022), the diffusion process encompasses only the state transitions as described by

$$x_{t_{i+1}}(\tau) := (s_k, s_{k+1}, \ldots, s_{k+H-1})_{t_{i+1}}. \tag{1}$$

In this notation, k indicates the timestep of a state within a trajectory τ, H represents the planning horizon, and t_{i+1} is the timestep in the diffusion sequence. Consequently, $x_{t_{i+1}}(\tau)$ is defined as a noisy sequence of states, represented as a two-dimensional array where each column corresponds to a different timestep of the trajectory.

To derive actions from the states generated by the diffusion model, we employ an inverse dynamics model (Agrawal et al., 2016; Pathak et al., 2018), denoted as h_φ, trained using the same dataset as the diffusion model. The combined training of the diffusion model (denoted by D_ϕ) and the inverse dynamics model is conducted using the following loss:

$$\begin{aligned} \mathcal{L}(\phi, \varphi) :=& \mathbb{E}_{\sigma\sim p_{train}, \tau\sim\mathcal{D}, n\sim\mathcal{N}(0,\sigma^2\mathrm{I}), \beta\sim\mathrm{Bern}(p)} \left[\| D_\phi(x_\sigma(\tau), (1-\beta)c(\tau) + \beta\emptyset, \sigma) - x_0(\tau) \|_2^2 \right] \\ &+ \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left[\left\| a - h_\varphi(s, s^{'}) \right\|_2^2 \right], \end{aligned} \tag{2}$$

where p_{train} is a log-normal distribution using the design choice from Karras et al. (2022), β is sampled from a Bernoulli distribution with probability p. Namely, the condition information $c(\tau)$ is ignored with probability p, which is manifested by the condition information being an empty set $\emptyset$.

We employ returns $R(\tau)$ under trajectories as the conditioning information $c(\tau)$, normalized such that $R(\tau) \in [0, 1]$. We map it into a latent variable $c \in \mathbb{R}^h$ using a multi-layer perceptron. In cases where $R(\tau) = \emptyset$, the components of c are set to zero. During the inference time, sampling trajectories with high returns corresponds to conditioning on $R(\tau) = 1$.

Guided Consistency Distillation. Incorporating classifier-free guidance is essential for synthesizing high-return trajectories. Considering the computational demands and potential for error accumulation associated with two-stage distillation methods (Meng et al., 2023), we opt for a one-stage guided distillation approach as proposed by Luo et al. (2023).

Algorithm 1 Consistency Distillation with guidance

Input: dataset $\mathcal{D}$, intial consistency model parameter θ, learning rate η, ODE solver $\Phi(\cdot, \cdot, \cdot; \phi)$, distance metric $d(\cdot, \cdot)$, EMA rate μ, noise schedule t_i, guidance schedule $[\omega_{min}, \omega_{max}]$.
$\theta^- \leftarrow \theta$
repeat
  Sample $(x, c) \sim \mathcal{D}$, $n \sim \mathcal{U}[1, N-1]$ and $\omega \sim [\omega_{min}, \omega_{max}]$
  Sample $x_{t_{n+1}} \sim \mathcal{N}(x; t_{n+1}^2 \mathrm{I})$
  $\hat{x}_{t_n}^{\phi,\omega} \leftarrow x_{t_{n+1}} + \left[(\omega+1)\Phi(x_{t_{n+1}}, c, t_{n+1}; \phi) - \omega\Phi(x_{t_{n+1}}, \emptyset, t_{n+1}; \phi)\right]$
  $\mathcal{L}(\theta, \theta^-; \phi) \leftarrow d\left(f_\theta(x_{t_{n+1}}, \omega, c, t_{n+1}), f_{\theta^-}(\hat{x}_{t_n}^{\phi,\omega}, \omega, c, t_n)\right)$
  $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}(\theta, \theta^-; \phi)$
  $\theta^- \leftarrow \text{stopgrad}(\mu\theta^- + (1-\mu)\theta)$
until convergence

The consistency function $f_\theta : (x_t, \omega, c, t) \rightarrow x_0$ is parameterized to transform state x_t at time t directly into the original state x_0. We parameterize f_θ in the same way as Song et al. (2023), except that we consider the influences of guidance scale ω and conditioning variable c:

$$f_\theta(x, \omega, c, t) = c_{skip}(t)x + c_{out}(t)F_\theta(x, \omega, c, t), \tag{3}$$

where F_θ is a free-form neural network with an output that matches the dimensionality of x, $c_{skip}(\epsilon) = 1$ and $c_{out}(\epsilon) = 0$ so that f_θ satisfies boundary condition $f_\theta(x, \omega, c, \epsilon) \equiv x$. During the distillation process, the guidance scale ω and n are sampled uniformly from the intervals $[\omega_{min}, \omega_{max}]$ and $\{1, \cdots, N-1\}$, respectively. The trajectory and returns tuple (x, c) are sampled from the dataset. Then, $\hat{x}_{t_n}^{\phi,\omega}$ is estimated by employing an ODE solver Φ:

$$\hat{x}_{t_n}^{\phi,\omega} - x_{t_{n+1}} \approx \left[(\omega+1)\Phi(x_{t_{n+1}}, c, t_{n+1}; \phi) - \omega\Phi(x_{t_{n+1}}, \emptyset, t_{n+1}; \phi)\right]. \tag{4}$$

Finally, we minimize the consistency distillation loss (Song et al., 2023; Luo et al., 2023) used for guided distillation:

$$\mathcal{L}(\theta, \theta^-; \phi) = \mathbb{E}_{x,c,\omega,n}\left[d\left(f_\theta(x_{t_{n+1}}, \omega, c, t_{n+1}), f_{\theta^-}(\hat{x}_{t_n}^{\phi,\omega}, \omega, c, t_n)\right)\right], \tag{5}$$

where d is squared ℓ_2 distance $d(x, y) = \|(x-y)\|_2^2$.

The pseudo-code for guided consistency distillation adapted for trajectory generation is shown in Algorithm 1.

Consistency Model Inference. During the inference process, we first observe a state s in the environment and sample an initial trajectory x_T. Then, our consistency model, which conditioned on returns c, guidance scale ω and history of last C states observed, iteratively predicts the denoised trajectories from the noisy inputs $\hat{x}_{t_n}(\tau) \leftarrow x(\tau) + \sqrt{t_n^2 - \epsilon^2}z$ along the probability flow ODE

Algorithm 2 Planning with Consistency Model

Input: consistency model f_θ, inverse dynamics h_φ, guidance scale ω, history length C, condition c.
Initialize $h \leftarrow Queue(length = C)$, $t \leftarrow 0$.
while not *done* **do**
  Observe state s; $h.insert(s)$;
  Initialize $x(\tau) \leftarrow f_\theta(x_T, \omega, c, T)$, $x_T \sim \mathcal{N}(0, T^2\mathrm{I})$
  for $n = 1$ to $N - 1$ **do**
    $x(\tau)[:, length(h)] \leftarrow h$
    $\hat{x}_{t_n}(\tau) \leftarrow x(\tau) + \sqrt{t_n^2 - \epsilon^2} z$, $z \sim \mathcal{N}(0, \mathrm{I})$
    $x(\tau) \leftarrow f_\theta(\hat{x}_{t_n}(\tau), \omega, c, t_n)$
  end for
  Extract (s_k, s_{k+1}) from $x(\tau)$
  Execute $a_k = h_\varphi(s_k, s_{k+1})$
end while

trajectory at step $n \in [N]$, with Gaussian noise $z \sim \mathcal{N}(0, \mathrm{I})$. For single-step version of Consistency Inference, $\{t_n \mid n = 0, 1\} = \{\epsilon, T\}$. Finally, we extract states (s_k, s_{k+1}) from denoised trajectory and get the action a_k via our inverse dynamics models h_φ. The algorithm of Consistency Planning is provided in Algorithm 2 and visualized in Figure 1. For the architecture and implementation details, please refer to Appendix.

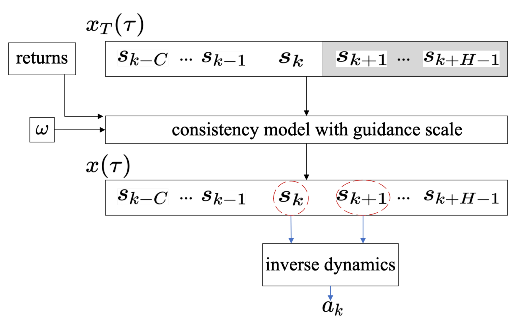

Figure 1: Consistency Planning. Given the current state s_k, conditioning variable and guidance scale ω, Consistency Planning generate a sequence of future states with planning horizon H. Then the inverse dynamics model is used to extract and execute the action a_k from s_k and s_{k+1}

5 Experiment

To evaluate the capabilities of the proposed consistency model in trajectory planning, we conduct experiments on Gym tasks (halfcheetah, hopper and walker2d) in D4RL benchmarks under offline RL settings.

We compare the performance of our method with those of both behavior-cloning methods, i.e., Consistency-BC (C-BC) (Ding & Jin, 2023), Diffusion-BC (D-BC) (Wang et al., 2022), actor-critic methods, i.e., Consistency-AC (C-AC) (Ding & Jin, 2023), Diffusion-QL (D-QL) (Wang et al., 2022) algorithms, and model-based methods, i.e., Diffuser (Janner et al., 2022), Decision-Diffuser (DD) (Ajay et al., 2022) in Table 1. For evaluation, results for our method correspond to the average over 150 planning seeds. By default, our consistency model applies the number of denoising steps N =

2 with saturated performance on most tasks, while the diffusion policy uses $N = 5$ (Wang et al., 2022), Diffuser and Decision Diffuser uses $N = 20$ and $N = 40$, respectively (Janner et al., 2022; Ajay et al., 2022).

Table 1 shows that although our method achieves a slightly lower average score (82.2) than Diffusion-QL (87.9) and Consistency-AC (85.1), as a model-based planning model, it outperforms its diffusion counterparts, i.e., Diffuser (75.3) and Decision Diffuser (81.8), with the reduction of denoising steps in the inference stages.

Dataset	**BC**	**Diffuser**	**DD**	**D-BC**	**C-BC**	**D-QL**	**C-AC**	**Ours**
Halfcheetah-me	55.2	79.8	90.6	90.8	32.7	96.8	84.3	94.0 ± 1.3
Hopper-me	52.5	107.2	111.8	107.6	90.6	111.1	100.4	107.5 ± 1.8
Walker2d-me	107.5	108.4	108.8	108.9	110.4	110.1	110.4	109.8 ± 0.5
Halfcheetah-m	42.6	44.2	49.1	45.4	31.0	51.1	69.1	46.8 ± 1.2
Hopper-m	52.9	58.5	79.3	65.3	71.7	90.5	80.7	87.8 ± 1.6
Walker-m	75.3	79.7	82.5	81.2	83.1	87.0	83.1	80.5 ± 0.8
Halfcheetah-mr	36.6	42.2	39.3	41.7	34.4	47.8	58.7	40.6 ± 0.9
Hopper-mr	18.1	96.8	100	67.9	99.7	101.3	99.7	97.8 ± 0.8
Walker2d-mr	26.0	61.2	75	77.5	73.3	95.5	79.5	75.3 ± 1.1
Average	51.9	75.3	81.8	76.3	69.7	87.9	85.1	82.2

Table 1: The average scores of vanilla BC (with Gaussian), Diffuser, Decision Diffuser, Diffusion-BC, Consistency-BC, Diffusion-QL, Consistency-AC and our method on D4RL Gym tasks are shown, with standard deviation reported for Consistency Planning. All results are quoted from Ding & Jin (2023) and Ajay et al. (2022).

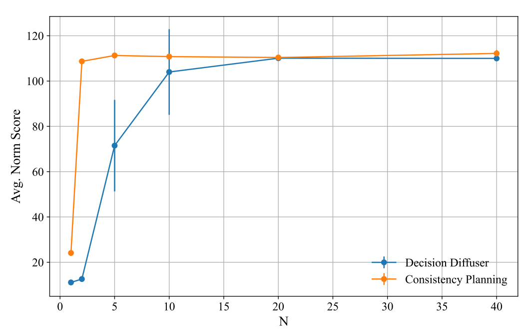

Figure 2: Comparison of average norm score vs. N for Decision Diffuser and Consistency Planning on the task hopper-medium-expert-v2

To assess the computational efficiency of Consistency Planning and its diffusion model counterparts, Decision Diffuser, we conduct experiments to measure inference time (ms per sample) in the hopper-medium-expert-v2 environment in our server. The results in Figure 2 show that $N = 2$ for Consistency-Planning, and $N = 20$ for Decision Diffuser, are the values where each algorithm achieves its saturated performance. The mean and standard deviation of results are calculated over five random seeds. We claim our model have achieved more than 12-fold increase in speed without any loss in performance, with more detailed information concerned the inference time v.s. N shown in Appendix.

6 Conclusion

By combining the score-based diffusion model proposed by Karras et al. (2022), one-stage guided distillation (Luo et al., 2023), and conditional model-based generative model for sequential decision making (Ajay et al., 2022), the consistency model in this paper achieves comparable performance in gym tasks with its diffusion model counterparts, Diffuser and Decision Diffuser, and obtains a significant speedup during inference in offline settings. Future work should include: 1) combining improved techniques in training consistency models (Song & Dhariwal, 2023), such as designing a changing weighting function and noise schedule more suitable for reinforcement learning scenarios; 2) combining the consistency inference process with changing guidance schedule (Ma et al., 2023) to improve the quality of trajectory sampling.

References

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *Advances in neural information processing systems*, 29, 2016.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Eloi Alonso, Adam Jelley, Anssi Kanervisto, and Tim Pearce. Diffusion world models. 2023.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. *arXiv preprint arXiv:2309.16984*, 2023.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

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

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, and Richard Zemel. Learning the stein discrepancy for training and evaluating energy-based models without sampling. In *International Conference on Machine Learning*, pp. 3732–3747. PMLR, 2020.

Haoran He, Chenjia Bai, Kang Xu, Zhuoran Yang, Weinan Zhang, Dong Wang, Bin Zhao, and Xuelong Li. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. *Advances in neural information processing systems*, 36, 2024.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Matthew Thomas Jackson, Michael Tryfan Matthews, Cong Lu, Benjamin Ellis, Shimon Whiteson, and Jakob Foerster. Policy-guided diffusion. *arXiv preprint arXiv:2404.06356*, 2024.

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

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

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

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

Cong Lu, Philip Ball, Yee Whye Teh, and Jack Parker-Holder. Synthetic experience replay. *Advances in Neural Information Processing Systems*, 36, 2024.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Jiajun Ma, Tianyang Hu, Wenjia Wang, and Jiacheng Sun. Elucidating the design space of classifier-guided diffusion generation. *arXiv preprint arXiv:2310.11311*, 2023.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14297–14306, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. *Advances in Neural Information Processing Systems*, 32, 2019.

Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 2050–2053, 2018.

Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*, 2023.

Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. *arXiv preprint arXiv:2004.00784*, 2020.

Marc Rigter, Jun Yamada, and Ingmar Posner. World models via policy-guided trajectory diffusion. *arXiv preprint arXiv:2312.08533*, 2023.

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

Laura Smith, J Chase Kew, Tianyu Li, Linda Luu, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Learning and adapting agile locomotion skills by transferring experience. *arXiv preprint arXiv:2304.09834*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

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

Appendix

The results in Table 2 show the relationship between the computational time, denoised steps N and corresponding performance of Consistency Planning and Decision Diffuser on the task hopper-medium-expert-v2. Each cell contains the mean and standard deviation over 5 random seeds. As demonstrated in Table 2, we achieved more than 12-fold increase in speed without any loss in performance with N=2 for Consisitency Planning and N=20 for Decision Diffuser.

Method	N	Inference Time (ms per sample)	Avg. Norm Score
Decision Diffuser	40	837.6 ± 8.4	110.0 ± 0.4
	20	427.7 ± 3.9	110.1 ± 0.5
	10	216.8 ± 1.2	104.0 ± 18.9
	5	107.3 ± 0.5	71.5 ± 20.2
	2	44.5 ± 0.3	12.6 ± 0.5
	1	23.1 ± 0.4	11.1 ± 0.5
Consistency Planning	40	752.1 ± 2.0	112.2 ± 1.5
	20	331.9 ± 1.9	110.4 ± 0.3
	10	167.3 ± 0.9	110.8 ± 1.0
	5	80.58 ± 0.7	111.3 ± 0.4
	2	33.2 ± 0.3	108.7 ± 0.9
	1	16.8 ± 0.2	24.1 ± 0.7

Table 2: Comparison of computational time for Decision Diffuser and Consistency Planning on the task hopper-medium-expert-v2 (Ajay et al., 2022).

In the next section, we describe various architectural and hyperparameter details:

- We represent the consistency model and diffusion model using the structure of Song et al. (2023), the inverse dynamics h_φ using the structure of Ajay et al. (2022).
- We train diffusion model using the same learning rate in Karras et al. (2022) and batch size of 512 for $2e5$ train steps.

- We choose the probability p of removing the conditioning information to be 0.25.
- We use $N = 2$ for consistency inference.
- We use a planning horizon H of 32, context length C of 8 in all the D4RL gym tasks.
- We use a guidance scale $\omega_{max} = 1, \omega_{min} = 0$ in guided consistency distillation.