Double Policy Estimation for Importance Sampling in Sequence Modeling Based Reinforcement Learning

Hanhan Zhou
The George Washington University
hanhan@gwu.edu

Tian Lan
The George Washington University
tlan@gwu.edu

Vaneet Aggarwal
Purdue University
vaneet@purdue.edu

Abstract

Offline reinforcement learning aims to utilize datasets of previously gathered environment-action interaction records to learn a policy without access to the real environment. Recent work has shown that offline reinforcement learning can be formulated as a sequence modeling problem and solved via supervised learning with approaches such as decision transformer. While these sequence-based methods achieve competitive results over return-to-go methods, especially on tasks that require longer episodes or with scarce rewards, importance sampling is not considered to correct the policy bias when dealing with off-policy data, mainly due to the absence of behavior policy and the use of deterministic evaluation policies. To this end, we propose an RL algorithm that blends offline sequence modeling and offline reinforcement learning with Double Policy Estimation (DPE) in a unified framework with statistically proven properties on variance reduction. We validate our method in multiple tasks of OpenAI Gym with D4RL benchmarks. Our method brings performance improvements on selected methods and outperforms state-of-the-art baselines in several tasks, demonstrating the advantages of enabling double policy estimation for sequence-modeled reinforcement learning.

1 Introduction

Offline reinforcement learning (RL) algorithms provide a promising approach for sequential decision-making tasks without the need for online interactions with an environment[Mei et al., 2023a, Zhou et al., 2022, Chen et al., 2021a]. This approach is particularly appealing when online interactions are costly or when there is an abundance of offline experiences available. Recent works have demonstrated that generative models [Chen et al., 2020, Brown et al., 2020, Radford et al., 2018, Zhou et al., 2021, He et al., 2023, Chen et al., 2023a,b] that are widely used in language and vision tasks can be applied to maximize the likelihood of trajectories in an offline dataset without temporal difference learning [Janner et al., 2021], notably, Decision Transformer (DT) [Chen et al., 2021b], which uses the transformer architecture [Vaswani et al., 2017] for decision-making. Such a pertaining paradigm in a supervised learning manner for RL can be considered known as Reinforcement learning via Supervised Learning (RvS) [Emmons et al., 2021, Schmidhuber, 2019, Srivastava et al., 2019]. Instead of learning a value-based algorithm for decision-making, RvS-based methods often consider the learning task as a prediction problem: to predict an action that will lead to a certain outcome or reward when given a sequence of past states and actions (e.g., using causal transformer architectures). These methods have gained significant attention due to their algorithmic and implementation simplicity while bringing a robust performance on several offline-RL benchmarks.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Learning an RvS policy π_e requires off-policy learning since we need to estimate the expected return of the learned policy π_e during training, from offline experiences/trajectories that are generated using a different behavior policy π_b. We note that online policy evaluation is usually expensive, risky, or even unethical for many real-world problems [Jiang and Li, 2016]. When the actual environment is not accessible, these trajectories sampled by π_b can be used to evaluate π_t, also known as off-policy evaluation (OPE) [Sutton and Barto, 2018]. An accurate OPE is crucial to evaluate and optimize a policy during training from offline datasets, the concept of importance sampling (IS) rectifies the discrepancy between the distributions of the behavior policy π_b and the evaluation policy π_e [Precup et al., 2000]. IS-based off-policy evaluation methods have also seen lots of interest recently, especially for short-horizon problems [Hirano et al., 2003, Murphy et al., 2001], including contextual bandits [Wang et al., 2017]. However, the application of IS to sequence modeling-based methods is difficult due to a number of challenges. The behavior policies for collecting experience/trajectory data are often not available, while the evaluation policies in RvS methods are typically deterministic, making reweighting different experience/trajectories inaccessible. Further, the variance of IS-based approaches tends to be too high to provide informative results, for long-horizon problems, since the variance of the product of importance weights may grow exponentially as the horizon goes long[Gottesman et al., 2019, Hanna et al., 2019, Zhou et al., 2023, Zhao et al., 2023].

In this paper, we study a problem that when given a dataset of trajectories sampled by a behavior policy and trajectories generated with sequence-modeling-based evaluation policy (in this paper we select Decision Transformer to demonstrate our approach), to estimate both behavior policy and target policy and then compute the importance sampling estimate which we call double policy estimation importance sampling. We further provide a theoretical analysis of the properties of such estimators and show that this double policy estimation will reduce the variance of the target policy learned.

Specifically, we propose to introduce an asymptotic estimation for both behavior policy π_b, which is used to sample and generate the dataset, and target evaluation policy, π_t, which is the policy we are in an attempt to learn and correct, as double policy estimation, to calculate the likelihood ratio for all state-action pairs in the off-policy data. Although it may seem that such an estimation would bring even worse performance as it introduces more uncertainties, recent research in several domains including multi-armed bandits [Li et al., 2015, Narita et al., 2019], Monte Carlo integration [Delyon and Portier, 2016], and causal inference [Hirano et al., 2003] has shown this estimating behavior could potentially improve the mean squared error of importance sampling policy evaluation which partially motivates this design. Another direct motivation is that specifically for many sequence modeling models based methods using generative models like decision transformer in an offline reinforcement an often scenario is that both π_b and π_t are inaccessible, which promotes a design for double policy estimation. In this paper we prove that DPE can statistically lower the mean squared error of importance sampling OPE with lower variance. We implement the proposed DPE on D4RL environments and compare DPE with SOTA baselines including DT [Chen et al., 2021b], RvS [Emmons et al., 2021], CQL [Kumar et al., 2020], BEAR [Kumar et al., 2019], UWAC [Wu et al., 2021], BC [Wu et al., 2019], and IQL [Kostrikov et al., 2021]. We empirically found double policy estimation based on importance sampling also brings an improvement to the off-policy evaluation of the D4RL environment, where DPE achieves better performance than the original decision transformer on almost all datasets and outperforms the state-of-the-art baselines over several datasets with further analysis discussing the effects and properties of the proposed double policy estimator.

2 Background

2.1 Markov Decision Process and Sequence-Based Method in Reinforcement Learning

We assume that the environment is a Markov decision process with a finite horizon and episodic nature, where the state space is denoted as $\mathcal{S}$, the action space as $\mathcal{A}$, and the environment possesses transition probabilities represented by P, a reward function denoted as R, a horizon length of H, a discount factor of γ, and initial state distribution of d_0 [Puterman, 2014, Li et al., 2015, Chen et al., 2023c]. A policy, denoted as π, is considered Markovian if it maps the current state to a probability distribution over actions. In contrast, a policy is classified as non-Markovian if its action distribution is dependent on past actions or states [Agarwal et al., 2022, Chen and Lan, 2023]. We assume $\mathcal{S}$ and $\mathcal{A}$ are finite for simplicity and probability distributions are probability mass functions.

In off-policy policy evaluation, we are given a fixed evaluation policy, π_e, and a data set of m trajectories and the policies that generated them: $\mathcal{D}\{\omega^i, \pi_b^{(i)}\}_{i=1}^m$ where $\omega^i \sim \pi_b^{(i)}$. We assume that $\forall\{\omega^i, \pi_b^{(i)}\} \in \mathcal{D}$, $\pi_b^{(i)}$ is Markovian, i.e., actions in $\mathcal{D}$ are independent of past states and actions gave the immediately preceding state. Sequence-based methods in reinforcement learning, which is trained in reinforcement learning via supervised learning (RvS) manner such as Decision Transformer, train a model using supervised learning on a dataset with respect to trajectories to predict $p_{\mathcal{D}(a|s,R)}$, i.e., given a cumulative reward $R = \sum_t \gamma^t r_t$ to predict the probability of next action conditioning the current state. Then at the deployment stage, the model takes actions conditioned on a desired target return value. Our goal is to design an off-policy estimator that takes $\mathcal{D}$ as input and estimates both behavior policy π_b and evaluation policy π_e for enabling importance sampling in sequence modeling methods.

Decision Transformer processes a trajectory $\boldsymbol{\omega}$ as a sequence consisting of 3 types of input to be tokenized: the states, actions selected, and the return-to-go. Specifically, it learns a deterministic model $\pi_{\text{DT}(a_t|\mathbf{a}_{-K,t},\mathbf{s}_{-K,t},\mathbf{r}_{-K,t})}$ where $-K$ denotes the past K sequences and is trained to predict the action token at timestamp t. During the evaluation, DT is given a desired reward g_0 and the initial stage s_0 at the beginning and executes the action it generates. Once an action a_t is generated and then executed, the next state $s_{t+1} \sim P(\cdot|s_t, a_t)$ and reward $r_t = R(s_t, a_t)$ are observed, together with the return-to-go $g_{t+1} = r_t - g_t$: this new sequence will be appended to the previous input. The process is repeated until the terminal state. DT is then trained under standard l_2 loss as $\nabla_\theta J(\pi_{DT}) = \frac{1}{K}\sum_k \nabla_{\theta_{DT}}(a_k - \hat{a})^2$ in a supervised learning way.

2.2 Importance Sampling in Reinforcement Learning

Importance Sampling (IS) is a method for reweighting returns generated by a behavior policy π_b, such that they are unbiased returns from the evaluation policy. Assuming there is a family of sampling distributions, $p(x;\eta)$, with parameter η, that generates a random trajectory $\omega := (s_0, a_0, r_0, \cdots, s_{L-1}, a_{L-1}, r_{L-1})$ from $p(x;\eta_0)$, where $g(\omega) := \sum_{t=0}^{L-1}\gamma^t r_t$ be the discounted return with preliminary fixed η_0: an ordinary importance sampling (OIS) method provides an estimator of θ in the form of $\tilde{\theta} = \frac{1}{n}\sum_{i=1}^n \frac{f(x_i)}{p(x;\eta_0)}$. Then $\tilde{\theta}$ is an unbiased estimator of θ and $\tilde{\theta}$ is guaranteed to converge to θ as n goes to infinity according to the strong law of large numbers [Henmi et al., 2007].

In Monte Carlo problems with high-dimensional x, the target density $p(x)$ can be writing in a chain-like decomposition as $p(x) = p(x_1)\prod_{t=2}^d p(x_1|x_{1:t-1})$, where $x_{[1:t]} = (x_1, \cdots, x_t)$. With a set of m trajectories and the policy that generated each trajectory, the IS off-policy estimate of $v(\pi_e)$ is: $\text{IS}(\pi_e, \mathcal{D}) := \frac{1}{m}\sum_{i=1}^m g(\omega^{(i)})\prod_{t=0}^{L-1}\frac{\pi_e(a_t^{(i)}|s_t^{(i)})}{\pi_b(a_t^{(i)}|s_t^{(i)})}$.

We refer to the above as the ordinary importance sampling (OIS) estimator which uses the true behavior policy and refer to $\frac{\pi_e(a|s)}{\pi_b(a|s)}$ as the OIS weight for action a in state s. A standard approach to dealing with off-policy data is to correct the policy using importance sampling (IS) by applying cumulative density ratios $\nu_{0:t}$ [Kallus and Uehara, 2020, Hanna and Stone, 2018]. Then the policy gradient $Z(\theta)$ can be rewritten as an expectation over p_{π_b} and further estimated using an equivalent empirical expectation. The off-policy version of the classic REINFORCE algorithm [Williams, 1992] recognizes $Z(\theta) = \mathbb{E}[\nu_{0:H}\sum_{t=0}^H r_t \sum_{t=0}^H g_t]$ (recall that E is understood as $\mathbb{E}_{p_{\pi_b}}$) and uses the estimated policy gradient given by replacing $\mathbb{E}$ with $\mathbb{E}_n$. Later works obtained a policy gradient in terms of Q-function as $Z(\theta) = \mathbb{E}[\sum_{t=0}^H \nu_{0:t} g_t q_t]$ [Chen and Jiang, 2019, Mei et al., 2023b].

3 Related Work

3.1 Sequence-Based method in Reinforcement Learning

Much recent progress has been on formulating the offline decision-making procedure in offline reinforcement learning as a context-conditioned sequence modeling problem [Janner et al., 2021, Chen et al., 2021b, Yang et al., 2023, Jin and Ataman, 2022, Yang et al., Xu et al., 2023, Xiao et al., 2022, Yang et al., 2020]. Compared to the temporal difference methods, these works consider a paradigm that utilizes predictive models to generate desired actions from the observation sequence and the task specification like a supervised learning problem [Schmidhuber, 2019, Srivastava et al., 2019, Emmons et al., 2021] rather than learning a Q-function or policy gradients. Specifically, the Decision Transformer model [Chen et al., 2021b] trains the transformer architecture [Vaswani et al., 2017] as a model-free context-conditioned policy that takes the encoded reward-to-go, state, and action sequence as input to predict the action for the next step, and the Trajectory Transformer [Janner et al., 2021] trains transformer that first discretizes each dimension of the input sequence and shows that beam search can be used to improve upon the model-free performance. Various attempts have also been made to improve transformers in multi-agent RL and other areas including meta RL, and multi-task RL[Chen et al., 2023d, Jin et al., 2022]. However, these works do not consider the importance of sampling for offline reinforcement learning. Our work extended this area with the proposed double policy estimation and further improved the asymptotic variance of the ordinary method using the true sampling distribution.

3.2 Importance Sampling in Reinforcement Learning

The use of off-policy samples within reinforcement learning is a popular research area [Silver et al., 2014, Levine and Koltun, 2013]. Many of them rely on OIS or variants of OIS to correct for bias. The use of importance sampling ensures unbiased estimates, but at the cost of considerable variance, as quantified by the ESS measure [Jie and Abbeel, 2010]. The problem of sampling error applies to any variant of importance sampling using OIS weights, e.g., weighted importance sampling and per-decision importance sampling [Precup et al., 2000], the doubly robust estimator [Jiang and Li, 2016], and the MAGIC estimator [Thomas and Brunskill, 2016]. On-policy Monte Carlo policy evaluation is also subject to sampling error, as it is a specific case of ordinary importance sampling where the behavior policy and the evaluation policy are identical[Mei et al., 2022]. Among these important sampling methods, [Hanna et al., 2019] is the closest work but considers estimated behavior policy where their behavior policy estimate comes from the same set of data used to compute the importance sampling estimate; while we estimate the behavior policy prior to the training phase from the dataset and estimate the target policy from data generated from the target policy.

4 Methodology

In this section, we present the primary focus of our work: double policy estimation (DPE) importance sampling that corrects for sampling error in sequence modeling-based reinforcement learning. The key idea is to obtain the maximum likelihood estimate of both behavior and evaluation policies $\hat{\pi}_b^{\eta}$ and $\hat{\pi}_t^{\psi}$ and use them for computing the DPE cumulative density ratio. We further analyze the theoretical properties of DPE and prove that it is guaranteed to reduce the asymptotic variance of policy parameters.

4.1 DPE for sequence modeling-based reinforcement learning

Let $\mathcal{D}$ be a set of off-policy trajectories of length $H+1$ collected by a behavior policy π_b, denoted by $\mathcal{D} = \{\omega_i,\ \forall i\}$ with each trajectory $\omega_i = \{(s_0{}^{(i)}, a_0{}^{(i)}, r_0{}^{(i)}, \cdots, s_H{}^{(i)}, a_H{}^{(i)}, r_H{}^{(i)})\}$. For known behavior policy π_b and evaluation policy π_e^{θ}, OIS leverages the cumulative density ratio $\nu_{0:t} = \prod_{k=0}^{t} v_k$ (with density ratio $v_k = \pi_e^{\theta}(a_k|s_k)/\pi_b(a_k|s_k)$) to reweight the policy scores $g_t = \nabla_\theta \log \pi_e^{\theta}(a_t|s_t)$, such that they are unbiased estimates of the evaluation policy π_e^{θ}. In the off-policy version of the classic REINFORCE algorithm [Williams, 1992], the policy gradient under OIS is recognized as $Z(\theta) = \mathbb{E}[\nu_{0:H} q_{0:H} \sum_{t=0}^{H} g_t]$, where $q_{t:H} = \sum_{s=t}^{H} r_s$ is the return-to-go from step t to step H in trajectory ω (generated from behavior policy π_b). OIS can be easily extended to its step-wise form [Deisenroth et al., 2013, Chen and Jiang, 2019] with $Z(\theta) = \mathbb{E}[\sum_{t=0}^{H} \nu_{0:t} q_{t:H} g_t]$. OIS has been commonly used in off-policy reinforcement learning.

We note that when RL is recast as an offline sequence modeling problem (such as Decision Transformer [Chen et al., 2021b] and RvS [Emmons et al., 2021]), it also relies on off-policy learning. However, there are three challenges preventing OIS from being directly applied to sequence modeling-based RL. First, offline RL datasets often do not provide the actual behavior policy for collecting trajectories, making it impossible to access π_b in importance sampling. Second, sequence modeling-based RL usually are trained using a transformer structure to represent evaluation policy and to generate deterministic action outputs [Chen et al., 2021b]. We need to extend them to stochastic policies to obtain π_e in importance sampling. Finally, Meanwhile, OIS is known to have a high variance [Rasmussen and Ghahramani, 2003], also known as high sampling error in importance sampling[Hanna et al., 2019]. Methods to reduce importance-sampling variance are needed for sequence modeling-based RL.

To this end, we propose two maximum likelihood estimators of (stochastic) behavior and evaluation policies in sequence modeling-based RL, denoted by $\hat{\pi}_b^\eta$ and $\hat{\pi}_e^\psi$. A baseline return b_t^ξ is further estimated (using a mean-square error loss) in sequence modeling-based RL and is leveraged to mitigate the variance in policy learning. Given a set $\mathcal{D}$ of m trajectories, the proposed DPE with respect to the off-policy version of classic REINFORCE algorithm [Williams, 1992] is defined as:

$$Z_{\text{DPE}}(\theta|\eta, \psi, \xi, \mathcal{D}) = \mathbb{E}\left[\left(q_{0:H} - b_0^\xi\right)\prod_{t=0}^{H}\frac{\pi_e^\psi(a_t|s_t)}{\pi_b^\eta(a_t|s_t)}\left(\sum_{t=0}^{H} g_t\right)\right] \tag{1}$$

DPE can also be applied to the step-wise form [Deisenroth et al., 2013, Chen and Jiang, 2019], by replacing the density ratio v_k with its estimator $\hat{v}_k = \pi_e^\theta(a_k|s_k)/\pi_b(a_k|s_k)$ and by subtracting the return baseline b_t^ξ, i.e.,

$$Z_{\text{DPE}}(\theta|\eta, \psi, \xi, \mathcal{D}) = \mathbb{E}\left[\sum_{t=0}^{H}(q_{t:H} - b_t^\xi)\hat{v}_{0:t} g_t\right]. \tag{2}$$

The key idea of our DPE estimator for importance sampling is to leverage the maximum likelihood estimate of behavior and evaluation policies, denoted by $\hat{\pi}_b^\eta$ and $\hat{\pi}_t^\psi$ respectively. We introduce the proposed maximum likelihood estimators for $\hat{\pi}_b^\eta$ and $\hat{\pi}_e^\psi$ and minimum-mean-square estimator for b^ξ as following:

Maximum likelihood estimator for behavior policy $\hat{\pi}_b^\eta$. We consider estimating the $\hat{\pi}_b$, with maximum likelihood as $\hat{\pi}_b^\eta := \text{argmax}_{\pi_b}\sum_{\omega\in\mathcal{D}}\sum_t \log\pi_b(a|\omega_{t-n:t})$, so that it could provide a behavior policy action probability estimation while the training of DT. Specifically, in this work, for policy network estimator we consider learning π_b from $\mathcal{D}$ as a Gaussian distribution over actions with mean and standard deviation estimated from a neural network.

Maximum likelihood estimator for target policy $\hat{\pi}_t^\psi$. One key insight in this paper is that when assuming a Gaussian policy for target policy estimation, the estimator would be minimizing the mean-square error of action predictions, thus it is identical to sequence modeling-based RL like DT with MSE loss where its variance is this MSE specifically to each timestep while training. When obtaining the target policy estimator, although for decision transformer π_b is often not directly available and $\pi_b(a|s, R)$ cannot be served as this estimator, also estimating an ongoing learning method might be unstable and inefficient, we point out that this weight at specific timestep t can be considered as a Gaussian distribution with a mean of $\hat{a}_t$ and variance of the corresponding MSE. We explain why this can serve as target policy estimation later in the main theorem in detail.

Minimum-mean-square estimator for baseline b^ξ. Since b^ξ is trained to predict return-to-go by minimizing loss $\sum_{i=1}^{m}\left[q_{t:H} - b_t^\xi\right]^2$. This can be easily incorporated into sequence modeling-based Reinforcement Learning like Decision Transformer.

Training sequence modeling based RL using DPE. We summarize the general architecture of the learning pipeline on Algorithm 1 of applying DPE to the sequence-modeling-based target policy (Decision Transformer). We first obtain an empirical estimator of the behavior policy π_b prior to the training of the Decision Transformer in a warm-up phase. Then during the training phase, we acquire the target policy estimator as a Gaussian distribution $\hat{a_t^\eta} \sim \mathcal{N}(\hat{a}_t, \sigma^2)$ where $\hat{a}_t$ is the mean generated from the decision transformer, $\hat{\sigma^2}$ is the MSE that serves as variance from the loss calculated at a specific timestamp.

4.2 Problem formulation and DPE Objective

In offline sequence modeling-based reinforcement learning, we are given a data set of m offline trajectories $\omega = \{(s_0, a_0, r_0...)\}$, and the behavior policy π that is collected them. We denote the trajectories that are generated by the decision transformer as $\hat{\omega} = \{(\hat{s_0}, \hat{a_0}, \hat{r_0}...)\}$

Note that DPE objective can also be written as :

$$\text{DPE} := \frac{1}{m}\sum_{i=1}^{n} q(h_t)\prod_{t=0}^{L-1}\frac{\hat{\pi}_t^{(i)}(a_t^{(i)}|s_t^{(i)})}{\hat{\pi}_b^{(i)}(a_t^{(i)}|s_t^{(i)})} = \frac{1}{m}\sum_{i=1}^{n}\frac{\hat{w}_{\pi_t}(h_t)}{\hat{w}_{\pi_b}(h_t)}q(h_t) \tag{3}$$

The variance of $\tilde{\theta}$ is given by $\delta^2(f)/n$, where $\delta^2 = \delta^2(f) = \int\{\frac{f(x)}{p(x;\eta_0)-\theta}\}^2 p(x;\eta_0)dx$, thus the distribution of $\sqrt{n}(\tilde{\theta}-\theta)$ converges to Normal distribution $\mathcal{N}(0,\delta^2)$ as n increases to infinity according to central limit theorem.

4.3 Theoretical Properties of DPE

We analyze the asymptotic properties of the maximum likelihood estimator of behavior policy $\pi_b^{\hat{\eta}}$ (with optimal parameters $\hat{\eta}$), the maximum likelihood estimator of evaluation policy $\pi_e^{\hat{\psi}}$ (with optimal parameters $\hat{\psi}$), and the minimum mean-square error estimators of baseline b_t^{ξ} (with optimal parameters $\hat{\xi}$). We show that these estimators are able to reduce the variance of policy gradient estimates Z_{DPE}. More precisely, for a given set of m off-policy trajectories $\mathcal{D} = \{\omega_i,\ \forall i\}$, we consider the gradient estimate Z_{DPE} with DPE (in both per-episode form as Eq. (1) and per-step form as Eq. (2)), i.e.,

$$Z_{\text{DPE}} = \frac{1}{m}\sum_{i=1}^{m}(q_{0:H}^{(i)} - b_0^{\hat{\xi}})\hat{v}_{0:H}^{(i)}\left(\sum_{t=0}^{H} g_t^{(i)}\right)\text{and}Z_{\text{DPE}} = \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H}(q_{t:H}^{(i)} - b_t^{\hat{\xi}})\hat{v}_{0:t}^{(i)}g_t^{(i)}. \tag{4}$$

We show that the variance $\text{Var}(Z_{\text{DPE}})$ using optimal estimators $\hat{\psi}$, $\hat{\eta}$ and $\hat{\xi}$ is lower than the variance $\text{Var}(Z_{\text{OIS}})$ using some ground truth ψ_0, η_0 and ξ_0.

We begin with recognizing that both per-episode and per-step DPE can be consolidated using a general form:

$$Z_{\text{DPE}} = \frac{1}{n}\sum_{i=1}^{n}\frac{f(\omega_i;\hat{\psi})[G(\omega_i) - b(\hat{\xi})]}{P(\omega_i;\hat{\eta})} \tag{5}$$

Next, we show a few lemmas demonstrating some properties of the estimators $\hat{\psi}$, $\hat{\eta}$, and $\hat{\xi}$ and then prove the variance reduction lemma.

Lemma 1. *Let $F_\eta = -\frac{1}{m}\sum_{i=1}^{m}\partial_\eta^2 \log P(\omega_i;\hat{\eta_0})$ be the Fisher Information Matrix. We have*

$$\sqrt{m}(\hat{\eta}-\eta_0) = \frac{1}{\sqrt{m}}F_\eta^{-1}\cdot\sum_{i=1}^{m}\partial_\eta \log P(\omega_i;\eta_0) + O(1) \tag{6}$$

Proof Sketch. Since $\hat{\eta}$ is the maximum likelihood estimator that optimizes $P(\omega_i;\eta)$, we have $\partial_\eta\sum_{i=1}^{m}\log P(\omega_i;\eta) = 0$ at $\eta = \hat{\eta}$. Expanding the left-hand side from $\eta = \eta_0$ toward $\eta = \hat{\eta}$, we have $0 = \sum_{i=1}^{m}\partial_\eta \log P(\omega_i;\eta_0) + \sum_{i=1}^{m}\partial_\eta^2 \log P(\omega_i;\hat{\eta_0})\cdot(\hat{\eta}-\eta_0) + o(||\hat{\eta}-\eta_0||_2)$, which yields the desired result by rearranging the terms and leveraging Fisher Information Matrix F_η.

Lemma 2. *Let $F_\xi = \frac{1}{m}\sum_{i=1}^{m}[\partial_\xi b(\xi))]^T\cdot\partial_\xi b(\xi))$. For linear baseline estimators $b(\xi)$, we have*

$$\sqrt{m}(\hat{\xi}-\xi_0) = \frac{1}{\sqrt{m}}F_\xi^{-1}\cdot\sum_{i=1}^{m}[G(\omega_i) - b(\xi_0)]\cdot\partial_\xi b(\xi_0) + O(1) \tag{7}$$

Proof Sketch. Since $\hat{\xi}$ is the minimum mean-square-error estimator optimizing $\sum_{i=1}^{m}[G(\omega_i) - b(\xi)]^2$, we have $\partial_\xi\sum_{i=1}^{m}[G(\omega_i) - b(\xi)]^2 = 0$. Expanding the left-hand side from

$\xi = \xi_0$ toward $\xi = \hat{\xi}$, we have $0 = \partial_\xi \sum_{i=1}^m [G(\omega_i) - b(\xi_0)]^2 + \partial_\xi^2 \sum_{i=1}^m [G(\omega_i) - b(\xi)]^2 (\hat{\xi} - \xi_0) + o(||\hat{\xi} - \xi_0||_2$. It yields the desired result using the fact that $b(\xi)$ is linear (thus $\partial_\xi^2 b(\xi) = 0$) and using the definition of F_ξ.

Theorem 1: The asymptotic variance of Z_{DPE}, using optimal estimators $\hat{\psi}$, $\hat{\eta}$, and $\hat{\xi}$, is always less than that of Z_{OIS} using some ψ_0, η_0 and ξ_0, i.e.,

$$\mathrm{var}(Z_{\mathrm{DPE}}) = \mathrm{var}(Z_{\mathrm{OIS}}) - \mathrm{var}(V_A) - \mathrm{var}(V_B) \tag{8}$$

where V_A and V_B are projections of $\{\mu_i = f(\omega_i; \hat{\psi})[G(\omega_i) - b(\hat{\xi})]/P(\omega_i; \hat{\eta}),\ \forall i\}$ onto the row space of $S_\eta = \partial_\eta log P(\omega_i; \eta_0)$ and $S_\xi = \partial_\xi b(\xi_0)$, respectively.

Proof Sketch.

Step 1: Define auxiliary function $\mu_i = \mu(\omega_i; \eta, \psi, \xi) = \frac{f(\omega_i)[G(\omega_i) - b(\xi_0)]}{P(\omega_i; \eta)}$, such that Z_{DPE} (which is $\hat{\theta}$ in the notes with Z_{OIS} being θ) can be written in $\sum_{i=1}^n \mu(x_i; \theta, \xi, \eta) - \theta = 0$. Then expand it from η_0, ψ_0, ξ_0 to $\hat{\eta}, \hat{\psi}, \hat{\xi}$, to obtain $\sqrt{n}(\hat{\theta} - \theta) = \frac{1}{\sqrt{n}} \sum_{i=1}^n \mu(\omega_i; \theta, \xi_0, \eta_0) + E(\partial_\eta \mu)(\hat{\eta} - \eta) + E(\partial_\xi \mu)\sqrt{n}(\hat{\xi} - \xi) + O(1)$.

Step 2: Rearranging the terms, plugging in Lemma 1 and Lemma 2, and using the fact of $\sum_{i=1}^n S_\eta{}' F_\eta^{-1} S_\eta = 1$ and $\sum_{i=1}^n w_i^2 S_\xi{}' F_\xi^{-1} S_\xi = 1$, we obtain the equation below, where define S_ξ and S_eta here. Note that we use weights $w_i = 1$ throughout the proof.

Step 3, Recognize that S_ξ and S_η are orthogonal. The two terms in C (i.e., A and B) can be viewed as projecting μ_i onto orthogonal row spaces of S_ξ and S_η, respectively. Define these as V_A and V_B The first term on the right hand side in $\sqrt{n}(\hat{\theta} - \theta) = \frac{1}{\sqrt{n}} \sum_{i=1}^n \{\mu_i - \underbrace{E(\mu_i S_\eta{}') F_\eta^{-1} \cdot S_\eta}_{V_A} - \underbrace{E(\mu_i S_\xi{}') \cdot w_i^2 F_\xi^{-1}}_{V_B}\} + O(1)$ is indeed OIS since $\sqrt{n}(\hat{\theta} - \theta) = \frac{1}{\sqrt{n}} \sum_{i=1}^n \mu_i$. Then from Pythagorean relationship, we prove $\mathrm{var}(Z_{\mathrm{DPE}}) = \mathrm{var}(Z_{\mathrm{OIS}}) - \mathrm{var}(V_A) - \mathrm{var}(V_B)$. The theorem shows that the use of the DPE estimator always reduces the asymptotic variance of the estimator of OIS.

5 Experiments

In this section, we present an empirical study of applying Double Policy Estimator on Decision Transformer to verify the feasibility and effectiveness of our proposed method. We evaluate the performance of our proposed algorithm on the continuous control tasks from the D4RL benchmark and compare it with several popular SOTA baselines. Furthermore, we analyze some critical properties to confirm the rationality of our motivation.

5.1 Experiment Setup

We empirically evaluate the performance of our proposed algorithm on the **Gym Locomotion** v2: a series of continuous control tasks consisting of `HalfCheetah, Hopper, and Walker2d` datasets from the D4RL offline reinforcement learning benchmark [Fu et al., 2020] with medium, medium-replay, and medium-expert datasets which include mixed and suboptimal trajectories. Specifically, `Medium` dataset includes 1 million timesteps generated by a "medium" policy that achieves approximately one-third of the score of an expert policy; `Medium-Replay` includes 25k-400k timesteps that are gathered from the replay buffer of an agent trained to the performance of a medium policy; `Medium-Expert` includes 1 million timesteps generated by the medium policy and then concatenated with 1 million timesteps generated by an expert policy.

5.2 Baseline Selection

We compare our proposed algorithm to the following SOTA methods, where they aim to tackle the current challenges in offline reinforcement learning from different perspectives: Decision Transformer (DT) [Chen et al., 2021b], reward-conditioned behavioral cloning (RvS) [Emmons et al., 2021], Conservative Q-Learning (CQL) [Kumar et al., 2020], BEAR [Kumar et al., 2019], UWAC [Wu et al., 2021], behavior cloning (BC), and Implicit Q Learning (IQL)[Kostrikov et al., 2021]. CQL and IQL represent the state-of-the-art in model-free offline RL; RvS and DT represent the state-of-the-art in sequence-modeling-based supervised learning.

5.3 DPE weights implementation

Note when proposing double policy estimation, there is no specific limitation on how π_b and π_t are estimated and how DPE weights are calculated. In this empirical section, we consider the following as one possible implementation: (1) We first apply CQL to train a neural network that generates mean and variances for Gaussian distributions as maximum likelihood estimation to obtain the estimated behavior policy $\hat{\pi}_b$. (2) Then for each trajectory ω_i we can calculate the estimated behavior weights as $\hat{w}_i^{\pi_b} = \hat{\pi}_b(a_i|\omega_i)$ (3) Next we train DT using l_2 loss for updating each timestep, but we record the MSE $(a_i - \hat{a_i})^2$ as the variance, and $\hat{a_i}$ as the mean for the Gaussian distribution, i.e. $\mathcal{N}(a_i, (a_i - \hat{a_i})^2)$ as target policy estimation. (4) There are multiple ways to calculate these target weights, e.g. cumulative distribution function (CDF): $P(a_i - \beta < \hat{a_i} \leq a_i + \beta)$ where β is a probability offset, or probability density function (PDF). In this empirical result, we consider using exponentiated clipped log-likelihood: $\exp(l_a(\hat{a}, (a_i - \hat{a_i})^2))$ with $l_{\hat{a}}$ clipped at 0.05 and 0.995.

5.4 General Performance

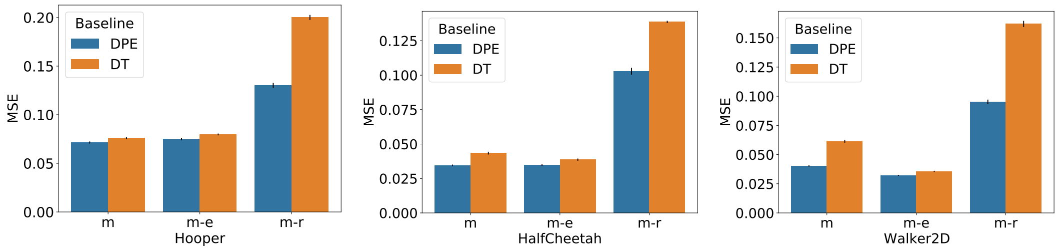
Dataset	Environment	DPE	DT	RvS	CQL	BEAR	UWAC	BC	IQL
medium	HalfCheetah	45.4±0.3	42.6±0.1	41.6	44.4	41.7	42.2	43.1	**47.4**
	Hopper	**69.8±1.9**	67.6±1.0	60.2	58.8	52.1	50.9	63.9	66.3
	Walker	77.9±0.8	74.0±1.4	71.7	**79.2**	59.1	75.4	77.3	78.3
medium-replay	HalfCheetah	40.5±1.5	36.6±0.8	38.0	**46.2**	38.6	35.9	4.3	44.2
	Hopper	**94.6±0.7**	79.4±7.0	73.5	48.6	33.7	25.3	30.9	94.5
	Walker	**83.5±1.2**	66.6±3.0	60.6	26.7	19.2	23.6	36.9	73.9
medium-expert	HalfCheetah	82.5±5.8	**87.8±2.6**	92.2	62.4	53.4	42.7	59.9	86.7
	Hopper	**108.2±1.6**	107.6±1.8	101.7	104.6	96.3	44.9	79.6	91.5
	Walker	93.7±6.2	108.1±0.2	106.0	108.1	40.1	96.5	36.6	**109.6**
average		**77.34**	74.60	71.72	64.33	48.24	48.60	48.06	76.93

Table 1: Overall performance of the normalized score of selected baselines on D4RL benchmark. All results are evaluated on 'v2' environments and datasets.

We first evaluate and compare the performance of the proposed method with all selected baselines in terms of average reward in Table 1, where 0 represents a random policy and 100 represents an expert policy, with reward normalized per [Fu et al., 2020]. All results are averaged over 3 different seeds over the final 10 evaluations, we put the full results including the error bar of all baselines in the appendix. Overall, we find DPE applied DT achieves better performance than the original decision transformer on almost all datasets, and outperforms the state-of-the-art baselines over several datasets. Especially, in 'medium-replay' datasets that include mixed optimal and sub-optimal trajectories, our method could bring a significant advancement in terms of reward. The finding that our proposed method attains competitive results stands in contrast to Decision Transformer which emphasizes the direct improvements brought by applying double policy estimation.

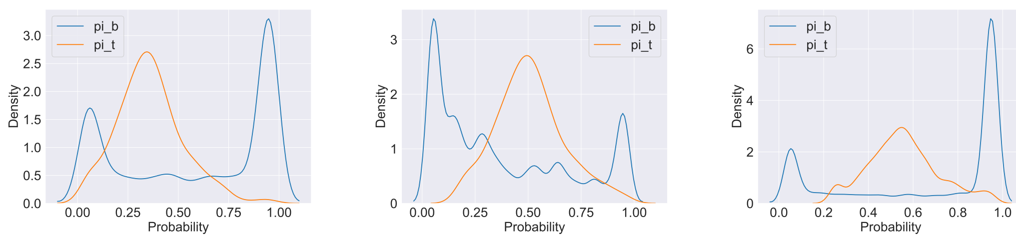
5.5 Discussions

To demonstrate the actual effectiveness of reducing the variance, we also record the MSE from the final evaluation stage of both DPE and DT for off-policy evaluation in Fig. 2, the results show that using DPE weights could bring a generally lower MSE on all environments selected compared to DT, validating our efficiency on variance reduction. To visualize the source of effectiveness in the double importance weights estimation we record the distribution of π_b and π_t on the 'hopper' environment and provide a kernel density estimate plot in Fig. 3. The drastic difference from the two distributions could mean that the behavior policy estimated are acting as a correction weight to offset the probability sampling from the target policy distribution, leading to improved performance and reduced variance. As an example, an occasional sub-optimal trajectory that the target trajectory



(a) Hopper-medium (b) HalfCheetah-medium (c) Walker2D-medium

Figure 1: MSE comparison with DT and DPE



(a) Hopper-medium (b) Hopper-medium-replay (c) Hopper-medium-expert

Figure 2: Comparing Kernel Density Estimate of estimated π_b and π_t on `Hopper` datasets.

learned with high probability could be corrected by the low probability from the estimated behavior policy, making this a low-weight trajectory to learn from.

5.6 Ablation Studies

According to the object of DPE, the estimation of π_b still determines the target policy weights. In this section, we evaluate and compare several different ways to calculate the exact probability generated from the estimated behavior distribution marking as CDF ± 0.1, CDF ± 0.2, PDF, clipped PDF, and demonstrate the results over `medium-replay` datasets in terms of MSE in Figure 3. We see that despite some cases, most of the settings are similar regarding their prospective MSE, indicating that when a proper estimation of this Gaussian distribution is obtained, their method of sampling probability is not a major concern. Nevertheless, we find that using a clipped PDF for behavior probability selection brings the lowest MSE in general.

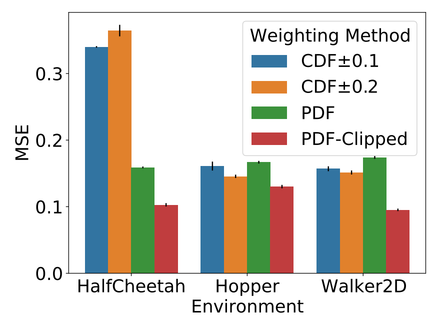


Figure 3: Ablations results on comparing different probability sampling methods on estimated π_b

6 Limitations and Social Impact

There are several opportunities for future work. First, our approach requires a warm-up phase prior to the training of the decision transformer to obtain the estimated behavior policy. Also, as RvS methods perform poorly in stochastic environments as pointed out in [Paster et al., 2022], the currently proposed method cannot resolve such issues. We only considered comparing the performance of applying DPE to decision transformer in this paper, although theoretically guaranteed to reduce the variance of sequence-modeling based decision making, the actual performance improvements on other designs remain further investigation.

7 Conclusion

In this paper, we present DPE, a double policy estimation for importance sampling methods that are proven statistically efficient for variance reduction for Off-Policy Evaluation in Sequence-Modeled Reinforcement Learning. Computing both the behavior policy estimate and target estimate from the same set of data allows DPE to correct for the sampling error inherent to importance sampling with the true behavior policy in the offline dataset. We evaluated DPE applied decision transformer across several benchmarks against popular works. We demonstrated its competitive performances while improving the evaluation results of the Decision Transformer, especially on the dataset filled with sub-optimal trajectories, and confirming the effect of variance reduction through MSE comparison. Finally, we studied the possible cause for such improvements by visualizing the density of the estimated target policy and behavior policy, providing potential insights on future designs.

References

Yongsheng Mei, Hanhan Zhou, Tian Lan, Guru Venkataramani, and Peng Wei. Mac-po: Multi-agent experience replay via collective priority optimization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 466–475, 2023a.

Hanhan Zhou, Tian Lan, and Vaneet Aggarwal. Pac: Assisted value factorization with counterfactual predictions in multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15757–15769. Curran Associates, Inc., 2022.

Jingdi Chen, Yimeng Wang, and Tian Lan. Bringing fairness to actor-critic reinforcement learning for network utility optimization. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021a.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Hanhan Zhou, Tian Lan, and Guru Venkataramani. Pt-vton: an image-based virtual try-on network with progressive pose attention transfer. *arXiv preprint arXiv:2111.12167*, 2021.

Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, Shigang Chen, Ronald Fick, Miles Medina, and Christine Angelini. A hierarchical spatial transformer for massive point samples in continuous space. *arXiv preprint arXiv:2311.04434*, 2023.

Yongchao Chen, Rujul Gandhi, Yang Zhang, and Chuchu Fan. Nl2tl: Transforming natural languages to temporal logics using large language models. *arXiv preprint arXiv:2305.07766*, 2023a.

Yongchao Chen, Jacob Arkin, Yang Zhang, Nicholas Roy, and Chuchu Fan. Autotamp: Autoregressive task and motion planning with llms as translators and checkers. *arXiv preprint arXiv:2306.06531*, 2023b.

Michael Janner, Qiyang Li, and Sergey Levine. Reinforcement learning as one big sequence modeling problem. In *Neural Information Processing Systems*, 2021.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

Juergen Schmidhuber. Reinforcement learning upside down: Don't predict rewards - just map them to actions. *ArXiv*, abs/1912.02875, 2019.

Rupesh Kumar Srivastava, Pranav Shyam, Filipe Wall Mutz, Wojciech Jaśkowski, and Jürgen Schmidhuber. Training agents using upside-down reinforcement learning. *ArXiv*, abs/1912.02877, 2019.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016.

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

Doina Precup, Richard S. Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. In *International Conference on Machine Learning*, 2000.

Keisuke Hirano, Guido W Imbens, and Geert Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.

Susan A Murphy, Mark J van der Laan, James M Robins, and Conduct Problems Prevention Research Group. Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, 96(456): 1410–1423, 2001.

Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudık. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pages 3589–3597. PMLR, 2017.

Omer Gottesman, Yao Liu, Scott Sussex, Emma Brunskill, and Finale Doshi-Velez. Combining parametric and nonparametric models for off-policy evaluation. In *International Conference on Machine Learning*, pages 2366–2375. PMLR, 2019.

Josiah Hanna, Scott Niekum, and Peter Stone. Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning*, pages 2605–2613. PMLR, 2019.

Hanhan Zhou, Tian Lan, and Vaneet Aggarwal. Value functions factorization with latent state information sharing in decentralized multi-agent policy gradients. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(5):1351–1361, 2023.

Pengyu Zhao, Zijian Jin, and Ning Cheng. An in-depth survey of large language model-based artificial intelligence agents. *arXiv preprint arXiv:2309.14365*, 2023.

Lihong Li, Rémi Munos, and Csaba Szepesvári. Toward minimax off-policy value estimation. In *Artificial Intelligence and Statistics*, pages 608–616. PMLR, 2015.

Yusuke Narita, Shota Yasui, and Kohei Yata. Efficient counterfactual learning from bandit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4634–4641, 2019.

Bernard Delyon and François Portier. Integral approximation by kernel smoothing. 2016.

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.

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

Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*, 2021.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

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

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Jingdi Chen, Lei Zhang, Joseph Riem, Gina Adam, Nathaniel D Bastian, and Tian Lan. Explainable learning-based intrusion detection supported by memristors. In *2023 IEEE Conference on Artificial Intelligence (CAI)*, pages 195–196. IEEE, 2023c.

Mridul Agarwal, Vaneet Aggarwal, and Tian Lan. Multi-objective reinforcement learning with non-linear scalarization. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 9–17, 2022.

Jingdi Chen and Tian Lan. Minimizing return gaps with discrete communications in decentralized pomdp. *arXiv preprint arXiv:2308.03358*, 2023.

Masayuki Henmi, Ryo Yoshida, and Shinto Eguchi. Importance sampling via the estimated sampler. *Biometrika*, 94(4):985–991, 2007.

Nathan Kallus and Masatoshi Uehara. Statistically efficient off-policy policy gradients. In *International Conference on Machine Learning*, pages 5089–5100. PMLR, 2020.

Josiah P Hanna and Peter Stone. Towards a data efficient off-policy policy gradient. In *AAAI Spring Symposia*, 2018.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32, 1992.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

Yongsheng Mei, Tian Lan, and Guru Venkataramani. Exploiting partial common information microstructure for multi-modal brain tumor segmentation. *arXiv preprint arXiv:2302.02521*, 2023b.

Yifan Yang, Chang Liu, and Zheng Zhang. Particle-based online bayesian sampling. *arXiv preprint arXiv:2302.14796*, 2023.

Zijian Jin and Duygu Ataman. Logographic information aids learning better representations for natural language inference. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 268–273, 2022.

Yifan Yang, Lin Chen, Pan Zhou, and Xiaofeng Ding. Vflh: A following-the-leader-history based algorithm for adaptive online convex optimization with stochastic constraints. *Available at SSRN 4040704*.

Zelin Xu, Tingsong Xiao, Wenchong He, Yu Wang, and Zhe Jiang. Spatial knowledge-infused hierarchical learning: An application in flood mapping on earth imagery. In *The 31st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*, 2023.

Tingsong Xiao, Lu Zeng, Xiaoshuang Shi, Xiaofeng Zhu, and Guorong Wu. Dual-graph learning convolutional networks for interpretable alzheimer's disease diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 406–415. Springer, 2022.

Yifan Yang, Jie Xu, Zichuan Xu, Pan Zhou, and Tie Qiu. Quantile context-aware social iot service big data recommendation with d2d communication. *IEEE Internet of Things Journal*, 7(6):5533–5548, 2020.

Yongchao Chen, Jacob Arkin, Yang Zhang, Nicholas Roy, and Chuchu Fan. Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? *arXiv preprint arXiv:2309.15943*, 2023d.

Zijia Jin, Xingyu Zhang, Mo Yu, and Lifu Huang. Probing script knowledge from pre-trained models. In *Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS)*, pages 87–93, 2022.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.

Sergey Levine and Vladlen Koltun. Guided policy search. In *International conference on machine learning*, pages 1–9. PMLR, 2013.

Tang Jie and Pieter Abbeel. On a connection between importance sampling and the likelihood ratio policy gradient. *Advances in Neural Information Processing Systems*, 23, 2010.

Philip S. Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. *ArXiv*, abs/1604.00923, 2016.

Yongsheng Mei, Tian Lan, Mahdi Imani, and Suresh Subramaniam. A bayesian optimization framework for finding local optima in expensive multi-modal functions. *arXiv preprint arXiv:2210.06635*, 2022.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

Carl Edward Rasmussen and Zoubin Ghahramani. Bayesian monte carlo. *Advances in neural information processing systems*, pages 505–512, 2003.

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

Keiran Paster, Sheila McIlraith, and Jimmy Ba. You can't count on luck: Why decision transformers fail in stochastic environments. *arXiv preprint arXiv:2205.15967*, 2022.

A Mathematical Details

A.1 Notations and Explanations

We use Table 1 to summarize the notations introduced in this paper and their corresponding explanations, more explanations can be found when first introduced in the main paper.

Notation	Definition
s	state space
a	action space
P	environment transition probabilities
r	reward, reward function
$\mathcal{D}$	offline dataset
ω	a trajectory
H	Horizon length
π_e	evaluation policy
π_b	behaviour policy
π_t	target policy
ν	density ratio
Z	importance weighed policy gradient
g	policy score
q	return-to-go
$\hat{\pi}_b$	estimated behavior policy
$\hat{\pi}_t$	estimated target policy
b^{η}	baseline predicted return-to-go
$\mathcal{N}(a, \sigma^2)$	Gaussian distribution, with mean of a and std of σ
Var	Variance Operator
$\hat{\xi}, \hat{\eta}, \hat{\psi}$	estimators

Table 1

A.2 DPE for sequence modeling-based reinforcement learning

Let $\mathcal{D}$ be a set of off-policy trajectories of length $H+1$ collected by a behavior policy π_b, denoted by $\mathcal{D} = \{\omega_i,\ \forall i\}$ with each trajectory $\omega_i = \{(s_0{}^{(i)}, a_0{}^{(i)}, r_0{}^{(i)}, \cdots, s_H{}^{(i)}, a_H{}^{(i)}, r_H{}^{(i)}))$.

We first denote the approximated behavior policy with a Gaussian distribution as $\pi_b = argmax_{\pi} P(\pi|\omega)$, and the approximated target policy as $\pi_t = \hat{a_t} + \hat{\sigma^2} * n_k$, where $\hat{a_t}$ and $\hat{\sigma^2}$is the mean and variance generated from the decision transformer, n_k is Gaussian noise.

A.3 Proof For Lemma 1

Lemma 1. *Let $F_\eta = -\frac{1}{m}\sum_{i=1}^{m} \partial_\eta^2 \log P(\omega_i; \hat{\eta_0})$ be the Fisher Information Matrix. We have*

$$\sqrt{m}(\hat{\eta} - \eta_0) = \frac{1}{\sqrt{m}} F_\eta^{-1} \cdot \sum_{i=1}^{m} \partial_\eta log P(\omega_i; \eta_0) + O(1) \tag{1}$$

Proof. Since $\hat{\eta}$ is the maximum likelihood estimator that optimizes $P(\omega_i; \eta)$, we have $\partial_\eta \sum_{i=1}^{m} \log P(\omega_i; \eta) = 0$ at $\eta = \hat{\eta}$:

$$
\begin{aligned}
0 &= \sum_{i=1}^{n} \partial_\eta logP(\omega_i;\eta_0) + \sum_{i=1}^{n} \partial_\eta^2 logP(\omega_i;\hat{\eta}_0)(\hat{\eta}-\eta_0) + O(||\hat{\eta}-\eta||_2) \\
&\text{Expanding the right-hand side from } \eta=\eta_0 \text{ toward } \eta=\hat{\eta}\text{, we have} \\
\sqrt{n}(\hat{\eta}-\eta_0) &= \frac{1}{\sqrt{n}}\{-\frac{1}{n}\partial_\eta^2 logP(\omega_i;\eta_0)(\hat{\eta}-\eta_0)\}^{-1} \cdot \sum_{i=1}^{n} \partial_\eta logP(\omega_i;\eta_0) + O(1) \\
\sqrt{n}(\hat{\eta}-\eta_0) &= \frac{1}{\sqrt{n}}\{-\frac{1}{n}\partial_\eta^2 logP(\omega_i;\eta_0)(\hat{\eta}-\eta_0)\}^{-1} \cdot \sum_{i=1}^{n} \partial_\eta logP(\omega_i;\eta_0) + O(1) \\
\sqrt{n}(\hat{\eta}-\eta_0) &= \frac{1}{\sqrt{n}}F_\eta^{-1} \cdot \sum_{i=1}^{n} \partial_\eta logP(\omega_i;\eta_0) + O(1) \qquad (2)
\end{aligned}
$$

which yields the desired result by rearranging the terms and leveraging the Fisher Information Matrix F_η.

A.4 Proof For Lemma 2

Lemma 2. *Let $F_\xi = \frac{1}{m}\sum_{i=1}^{m}[\partial_\xi b(\xi))]^T \cdot \partial_\xi b(\xi))$. For linear baseline estimators $b(\xi)$, we have*

$$\sqrt{m}(\hat{\xi}-\xi_0) = \frac{1}{\sqrt{m}}F_\xi^{-1} \cdot \sum_{i=1}^{m}[G(\omega_i) - b(\xi_0)] \cdot \partial_\xi b(\xi_0) + O(1) \qquad (3)$$

Proof. Since $\hat{\xi}$ is the minimum mean-square-error estimator optimizing $\sum_{i=1}^{m}\left[G(\omega_i) - b(\xi)\right]^2$, we have:

$$\partial_\xi \sum_{i=1}^{m}\left[G(\omega_i) - b(\xi)\right]^2 = 0 \qquad (4)$$

Expanding the right-hand side from $\xi = \xi_0$ toward $\xi = \hat{\xi}$, we have

$$
\begin{aligned}
0 &= \partial_\xi \sum_{i=1}^{m}\left[G(\omega_i) - b(\xi_0)\right]^2 + \partial_\xi^2 \sum_{i=1}^{m}\left[G(\omega_i) - b(\xi)\right]^2 (\hat{\xi}-\xi_0) + o(||\hat{\xi}-\xi_0||_2 \\
0 &= \sum_{i=1}^{n} \partial_\xi w_i^2 [G(\omega_i) - b(\xi)]^2 + \sum_{i=1}^{n} \partial_\xi^2 w_i^2 [G(\omega_i) - b(\xi)]^2 (\hat{\xi}-\xi)^2 + O(||(\hat{\xi}-\xi)||_2) \\
0 &= \sum_{i=1}^{n}(-I)\cdot[G(\omega_i) - b(\xi)]^2 w_i^2 \cdot \partial_\xi b(\xi_0) + \sum_{i=1}^{n}[w_i^2(\partial_\xi b(\xi_0))]^2 - I[G(\omega_i) - b(\xi_0)\cdot w_i^2 \cdot \partial_\xi^2 b(\xi_0)]^2(\hat{\xi}-\xi) + O(1) \\
\sqrt{n}(\hat{\xi}-\xi) &= \frac{1}{\sqrt{n}}\{\frac{1}{n}\sum_{i=1}^{n}(\partial_\xi^2 b(\xi_0))^2 w_i^2\}^{-1} \cdot \sum_{i=1}^{n}[G(\omega_i) - b(\xi_0)] \cdot w_i^2 \partial_\xi b(\xi_0) + O(1) \\
\sqrt{n}(\hat{\xi}-\xi) &= \frac{1}{\sqrt{n}}F_\xi^{-1} \cdot \sum_{i=1}^{n}[G(\omega_i) - b(\xi_0)] w_i^2 \partial_\xi b(\xi_0) \qquad (5)
\end{aligned}
$$

It yields the desired result using the fact that $b(\xi)$ is linear (thus $\partial_\xi^2 b(\xi) = 0$) and using the definition of F_ξ.

Proof Sketch. Since $\hat{\xi}$ is the minimum mean-square-error estimator optimizing $\sum_{i=1}^{m}\left[G(\omega_i) - b(\xi)\right]^2$, we have $\partial_\xi \sum_{i=1}^{m}\left[G(\omega_i) - b(\xi)\right]^2 = 0$. Expanding the left-hand side from $\xi = \xi_0$ toward $\xi = \hat{\xi}$, we have $0 = \partial_\xi \sum_{i=1}^{m}\left[G(\omega_i) - b(\xi_0)\right]^2 + \partial_\xi^2 \sum_{i=1}^{m}\left[G(\omega_i) - b(\xi)\right]^2 (\hat{\xi} - \xi_0) + o(||\hat{\xi}-\xi_0||_2$. It yields the desired result using the fact that $b(\xi)$ is linear (thus $\partial_\xi^2 b(\xi) = 0$) and using the definition of F_ξ.

A.5 Proof For Theorem 1

Theorem 1. *The asymptotic variance of Z_{DPE}, using optimal estimators $\hat{\psi}$, $\hat{\eta}$, and $\hat{\xi}$, is always less than that of Z_{OIS} using some ψ_0, η_0 and ξ_0, i.e.,*

$$var(Z_{\text{DPE}}) = var(Z_{\text{OIS}}) - var(V_A) - var(V_B) \tag{6}$$

where V_A and V_B are projections of $\{\mu_i = f(\omega_i;\hat{\psi})[G(\omega_i) - b(\hat{\xi})]/P(\omega_i;\hat{\eta}),\ \forall i\}$ onto the row space of $S_\eta = \partial_\eta log P(\omega_i;\eta_0)$ and $S_\xi = \partial_\xi b(\xi_0)$, respectively.

Proof. We define auxiliary function $\mu_i = \mu(\omega_i;\eta,\psi,\xi) = \frac{f(\omega_i)[G(\omega_i)-b(\xi_0)]}{P(\omega_i;\eta)}$, such that Z_{DPE} (which is $\hat{\theta}$ in the notes with Z_{OIS} being θ) can be written in

$$\sum_{i=1}^{n} \mu(\omega_i;\theta,\xi,\eta) - \theta = 0$$

Then expand it from η_0, ψ_0, ξ_0 to $\hat{\eta}, \hat{\psi}, \hat{\xi}$, we have

$$\begin{aligned}
0 &= \frac{1}{n}\mu(\omega_i;\theta,\xi_0,\eta_0) - (\hat{\theta}-\theta) + \frac{1}{n}\sum_{i=1}^{n}\partial_\eta\mu(\omega_i;\theta,\xi_0,\eta_0)(\eta-\hat{\eta}) + \frac{1}{n}\sum_{i=1}^{n}\partial_\xi\mu(\omega_i;\theta,\xi_0,\eta_0)(\xi-\hat{\xi}) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\partial_\xi\mu(\omega_i;\theta,\xi_0,\eta_0)(\hat{\xi}-\xi) + O(||\hat{\theta}-\theta||_2 + ||\hat{\eta}-\eta||_2 + ||\hat{\xi}-\xi||_2) \\
\sqrt{n}(\hat{\theta}-\theta) &= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mu(\omega_i;\theta,\xi_0,\eta_0) + E(\partial_\eta\mu)(\hat{\eta}-\eta) + E(\partial_\xi\mu)\sqrt{n}(\hat{\xi}-\xi) + O(1)
\end{aligned} \tag{7}$$

Rearranging the terms, plugging in Lemma 1 and Lemma 2, and using the fact of $\sum_{i=1}^{n} S_\eta{}' F_\eta^{-1} S_\eta = 1$ and $\sum_{i=1}^{n} w_i^2 S_\xi{}' F_\xi^{-1} S_\xi = 1$, we obtain the equation below, where define S_ξ and S_η here. Note that we use weights $w_i = 1$ throughout the proof. Recognize that S_ξ and S_η are orthogonal. The two terms in V_A and V_B can be viewed as projecting μ_i onto orthogonal row spaces of S_ξ and S_η, respectively. Define these as V_A and V_B

$$\begin{aligned}
\sqrt{n}(\hat{\theta}-\theta) &= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\{\mu_i - \underbrace{E(\mu_i S_\eta{}')F_\eta^{-1}\cdot S_\eta}_{V_A} - \underbrace{E(\mu_i S_\xi{}')\cdot w_i^2 F_\xi^{-1}\}}_{V_B} + O(1) \\
\sqrt{n}(\hat{\theta}-\theta) &= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mu_i + E\{-\frac{f(\omega_i)[G(\omega_i)-b(\xi_0)]}{P(\omega_i;\eta_0)}\cdot\frac{\partial_\eta P(\omega_i;\eta_0)}{P(\omega_i;\eta_0)}\cdot\sqrt{n}(\hat{\eta}-\eta_0)\} \\
&\quad + E\{-\frac{f(\omega_i)}{P(\omega_i;\eta_0)}\partial_\xi b(\xi_0)\}\cdot\sqrt{n}(\hat{\xi}-\xi) + O(1) \\
\sqrt{n}(\hat{\theta}-\theta) &= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mu_i - E(\mu_i\cdot s_\eta{}')\cdot\frac{1}{\sqrt{n}}F_\eta^{-1}\cdot\sum_{i=1}^{n}s_\eta \\
&\quad - E\{-\frac{f(\omega_i)}{P(\omega_i;\eta_0)}\partial_\xi b(\xi_0)\}\cdot\frac{1}{\sqrt{n}}F_\xi^{-1}\sum_{i=1}^{n}w_i^2[G(\omega_i)-b(\xi_0)]\partial_\xi b(\xi_0) + O(1) \\
&= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mu_i - \frac{1}{\sqrt{n}}E(\mu_i\cdot S_\eta{}')\cdot F_\eta^{-1}\cdot S_\eta - \frac{1}{\sqrt{n}}\sum_{i=1}^{n}E(\mu_i S_\eta{}')\cdot w_i^2 F_\eta^{-1}\cdot S_\xi + O(1+\frac{2}{\sqrt{n}})
\end{aligned} \tag{8}$$

Recall that

$$\begin{cases} \sum_{i=1}^{n} S_\eta{}' F_\eta^{-1} S_\eta = 1, \\ \sum_{i=1}^{n} w_i^2 S_\xi{}' F_\xi^{-1} S_\xi = 1 \end{cases}$$

We have:

$$\sqrt{n}(\hat{\theta}-\theta)=\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\{\mu_i-\underbrace{E(\mu_i S_{\eta}{}')F_{\eta}^{-1}\cdot S_{\eta}}_{V_A}-\underbrace{E(\mu_i S_{\xi}{}')\cdot w_i^2 F_{\xi}^{-1}\}}_{V_B}+O(1) \tag{9}$$

$$\tag{10}$$

We note that $S_{\xi}{}'$ and $S_{\eta}{}'$ are orthogonal, i.e.,

$$\sum_{i=1}^{n} S_{\eta}S_{\xi}=S_{\xi}\sum_{i=1}^{n}S_{\eta}=0$$

Recognize that S_{ξ} and S_{η} are orthogonal. The two terms in Eq.(10) can be viewed as projecting μ_i onto orthogonal row spaces of S_{ξ} and S_{η}, respectively. Define these as V_A and V_B The first term on the right hand side in

$$\sqrt{n}(\hat{\theta}-\theta)=\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\{\mu_i-\underbrace{E(\mu_i S_{\eta}{}')F_{\eta}^{-1}\cdot S_{\eta}}_{V_A}-\underbrace{E(\mu_i S_{\xi}{}')\cdot w_i^2 F_{\xi}^{-1}\}}_{V_B}+O(1)$$

is indeed OIS since $\sqrt{n}(\hat{\theta}-\theta)=\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\mu_i$. From Pythagorean relationship, we prove $\text{var}(Z_{\text{DPE}})=\text{var}(Z_{\text{OIS}})-\text{var}(V_A)-\text{var}(V_B)$. The theorem shows that the use of the DPE estimator always reduces the asymptotic variance of the estimator of OIS.

From Pythagorean relationship, we prove $\text{var}(Z_{\text{DPE}})=\text{var}(Z_{\text{OIS}})-\text{var}(V_A)-\text{var}(V_B)$. The theorem shows that the use of the DPE estimator always reduces the asymptotic variance of the estimator of OIS.

Algorithm 1 pseudocode for DPE

Initiate $\hat{\theta}$ for $\pi_{\theta}(a|s)$
for $k=0$ to $pretrain\_steps$ **do**
    Random Sample Trajectories: $\boldsymbol{\tau}\sim$ D
    Sample time index for each trajectory: $h\sim\tau_i[1,L]$
    Calculate Loss: $\mathcal{L}(\hat{\theta})=\sum_{s_t,a_t,h}\log\pi_{\theta}(a_t|s_t)$
    Update policy parameters: $\hat{\theta}=\hat{\theta}+\eta\nabla_{\theta}(\mathcal{L}(th\hat{e}ta))$
end for
initialize $\hat{\pi}_t$, Decision Transformer
for $k=0$ to max_train_steps **do**
    Random Sample Trajectories: $\boldsymbol{\tau}\sim$ D
    Sample time index for each trajectory: $h\sim\tau_i[1,L]$
    Generate Trajectories from DT: $\hat{a}=\text{DT}(R,s,a,t)$
    Calculate Loss for DT: $\mathcal{L}(\tilde{\theta})=\frac{1}{N}\sum_{i=1}^{N}(\hat{a_i}-a_i)^2$
    Estimate $\nabla_{\theta}J(\pi_t)=\sum_{i,t}\nabla_{\theta_t}\log\pi_t(a_{i,t}|s_{it})A((a_{i,t}|s_{it}))$
    Update Decision Transformer: $\theta_{\pi_t}=\theta_{\pi_t}+\eta\nabla_{\theta_t}J(\pi_{\theta_t})$
end for
Return Decision Transformer π_t

B Experimental Details

Code for experiments can be found on GitHub(in the supplementary material during review).

B.1 Implementation Details

Our code is based on the original Decision Transformer and CQL. We summarize the pseudocode for DPE training process in Algorithm 1. The hyperparameters used are shown below.

We compare with 2 additional baselines here in Table 3.

Table 2: Hyperparameters of DPE in experiments for D4RL Dataset.

Hyperparameter	**Value**
Number of layers	3
Number of attention heads	1
Embedding dimension	128
Nonlinearity function	ReLU
Batch size	64
Context length K	20 HalfCheetah, Hopper, Walker
Return-to-go conditioning	6000 for HalfCheetah 3600 for Hopper 5000 for Walker
Dropout	0.1
Learning rate	10^{-4}
Grad norm clip	0.25
Weight decay	10^{-4}
Learning rate decay	Linear warmup for first 10^5 training steps

Dataset	Environment	DPE	DT	RvS	CQL	BEAR	TD3+BC	MOPO	UWAC	BC	IQL
medium	HalfCheetah	45.4	42.6	41.6	44.4	41.7	48.3	73.1	42.2	43.1	47.4
	Hopper	69.8	67.5	60.2	58.8	52.1	59.3	38.3	50.9	63.9	66.3
	Walker	77.9	74.0	71.7	79.2	59.1	83.7	41.2	75.4	77.3	78.3
medium-replay	HalfCheetah	40.5	36.6	38.0	46.2	38.6	44.6	69.2	35.9	4.3	44.2
	Hopper	94.6	79.4	73.5	48.6	33.7	60.9	32.7	25.3	30.9	94.5
	Walker	83.5	66.6	60.6	26.7	19.2	81.8	73.7	23.6	36.9	73.9
medium-expert	HalfCheetah	82.5	89.0	92.2	62.4	53.4	90.7	70.3	42.7	59.9	86.7
	Hopper	108.2	107.6	101.7	104.6	96.3	98.0	60.6	44.9	79.6	91.5
	Walker	93.7	108.1	106.0	108.1	40.1	110.1	77.4	96.5	36.6	109.6
	average										

Table 3