Adaptive Two-Level Quasi-Monte Carlo for Soft Actor-Critic

Du Ouyang[*]
Department of Mathematical Sciences
Tsinghua University
oyd21@mails.tsinghua.edu.cn

Zhenpeng Shi[*]
Tsinghua-Berkeley Shenzhen Institute
Tsinghua University
shizp22@mails.tsinghua.edu.cn

Aodong Guo
School of Mathematics and Statistics
Wuhan University
diego97@whu.edu.cn

Huaze Tang
Tsinghua-Berkeley Shenzhen Institute
Tsinghua University
thz21@mails.tsinghua.edu.cn

Hejin Wang
Department of Mathematical Sciences
Tsinghua University
wanghj20@mails.tsinghua.edu.cn

Chao Wang
Tsinghua-Berkeley Shenzhen Institute
Tsinghua University
wangchao23@mails.tsinghua.edu.cn

Wenbo Ding[†]
Tsinghua-Berkeley Shenzhen Institute
Tsinghua University
ding.wenbo@sz.tsinghua.edu.cn

Abstract

In the framework of Actor-Critic, the policy gradient is often expressed in the form of an integral $\mathbb{E}\left[h(X)\right]$. To estimate this integral with better convergence results, the quasi-Monte Carlo (QMC) method can be used in conjunction with the maximum sample size of 2^M, and the resulting estimator $\widehat{I}_{2^M}^{\mathrm{QMC}}$ achieves an error rate of $O(2^{-M+\varepsilon})$ with an arbitrarily small $\varepsilon > 0$. However, such a large number of QMC points often results in a substantial computational cost. To address this issue, we propose an adaptive two-level quasi-Monte Carlo (ATQ) method for approximating $\mathbb{E}\left[h(X)\right]$ with much fewer samples than $\widehat{I}_{2^M}^{\mathrm{QMC}}$. The ATQ method comprises two levels: the base level and the stochastic level. The base level employs large sample sizes to increase accuracy in the unstable phase of learning, and shifts to small sample sizes to save costs once stability is achieved. Within the stochastic level, we randomize the number of samples to ensure that the ATQ method is an unbiased estimator of $\widehat{I}_{2^M}^{\mathrm{QMC}}$. Theoretically, for the sample size 2^b of the base level, the ATQ method converges to $\mathbb{E}\left[h(X)\right]$ at the rate of $O(2^{-b+\varepsilon})$ with an arbitrarily small $\varepsilon > 0$, which is better than the Monte Carlo (MC) rate $O(2^{-b/2})$. Experimentally, we compare the ATQ-based Soft Actor-Critic method with strong baselines in both online Mujoco environments and offline D4RL suboptimal datasets. Our approach achieves state-of-the-art performance, outperforming other on-policy and off-policy methods in most aforementioned online environments and offline datasets.

[*]These authors contributed equally.
[†]Corresponding author.

Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

1 Introduction

The Soft Actor-Critic (SAC), proposed by Haarnoja et al. [14], is an efficient reinforcement learning approach, where computing the policy gradient is a crucial component. Under some reasonable settings, the calculation of the policy gradient boils down to estimating an integral $\mathbb{E}\left[h(X)\right]$ (see the form of h in Section 3), where X is a d-dimensional standard Gaussian random vector. In traditional Actor-Critic (AC) [22] or SAC frameworks, this integral is typically estimated by single samples which, mathematically, leads to inaccuracy shortcomings. Regarding this issue, one can consider using the Monte Carlo (MC) [12] method to approximate the policy gradient. Arnold et al. [2] applied the randomized quasi-Monte Carlo (RQMC) [32] method to SAC. Leveraging the advantage of the RQMC method, their algorithm achieved promising results. However, they did not provide the convergence rate of their method. To numerically compute $\mathbb{E}\left[h(X)\right]$, one can use the QMC quadrature rule $\widehat{I}_{2^M}^{\mathrm{QMC}} := \frac{1}{2^M}\sum_{i=1}^{2^M} h\circ\Phi^{-1}(y_i)$, where $\{y_1, y_2, \dots\}$ is a low discrepancy sequence, $\circ$ is the composite operator and Φ^{-1} is the inverse cumulative distribution function of the standard Gaussian distribution acting on each component of y_j. Based on Ouyang et al. [33], we prove that $\widehat{I}_{2^M}^{\mathrm{QMC}}$ converges to $\mathbb{E}\left[h(X)\right]$ (policy gradient) at the rate of $O(2^{-M+\varepsilon})$ with an arbitrarily small $\varepsilon > 0$, which is faster than the MC rate $O(2^{-M/2})$ (see Theorem 1). Our theoretical findings validate the superiority of the QMC method as evidenced in the experimental results of Arnold et al. [2].

Theoretically and intuitively, to achieve optimal results, one selects the largest M such that 2^M is the maximum computational tolerance for sample size. Nevertheless, employing such an extensive number of QMC points introduces substantial computational cost. To address this issue, we propose an adaptive two-level quasi-Monte Carlo (ATQ) method, aimed at reducing computation while enhancing reinforcement learning performance.

Our ATQ method, inheriting the exploration and exploitation concept from SAC, consists of two levels: the base level and the stochastic level. The base level is dedicated to exploitation, using 2^b sample points, ensuring that the ATQ method converges to $\mathbb{E}\left[h(X)\right]$ at a rate of $O(2^{-b+\varepsilon})$. The stochastic level is for exploration, where we randomize the number of samples to render the ATQ method an unbiased estimator of $\widehat{I}_{2^M}^{\mathrm{QMC}}$ (see Theorem 1). Moreover, we adjust the b during the training period. A larger b is employed to increase accuracy in the unstable phase of learning, while a smaller b is utilized to reduce computational costs once training stabilizes (see Section 3 for more details). This approach substantially reduces the total number of samples used throughout training while ensuring improved reinforcement learning outcomes.

The ATQ method combines multilevel Monte Carlo and QMC methods. Whereas the classical randomized multilevel Monte Carlo method [11, 16, 39] employs N stochastic levels, our method simplifies this by utilizing just a single stochastic level. Such a one stochastic level approach is also present in [43]. In contrast, we incorporate a base level to adjust the convergence rate of ATQ method, and the distribution of our stochastic level varies in response to changes within the base level, thus achieving an unbiased estimator of $\widehat{I}_{2^M}^{\mathrm{QMC}}$.

Our contributions are threefold. First, we propose the ATQ method for estimating the policy gradient and theoretically prove that our ATQ method has a high order of convergence. Second, our method dynamically adjusts the number of samples at the base level, which results in the use of far fewer total samples, thereby reducing computational costs. Third, we experimentally validate the efficiency of the ATQ-based SAC, which outperforms other baselines in both offline and online settings.

2 Background

2.1 Quasi-Monte Carlo

The numerical computation of integrals in various fields such as statistics, financial engineering, machine learning, and reinforcement learning often revolves around expectations. There are two methodologies for numerically solving these integrals–Monte Carlo (MC) [12] and quasi-Monte Carlo (QMC) [32]. MC uses random sampling and has a convergence rate of $O(n^{-1/2})$ with n quadrature points, while QMC utilizes low discrepancy sequences, such as Faure sequence, Sobol' sequence and Halton sequence (see [3, 32, 37]), yielding a faster convergence rate of $O(n^{-1+\varepsilon})$ with an arbitrarily small $\varepsilon > 0$. The convergence rate for QMC is on the Koksma-Hlawka inequality [19],

which relates the integration error to the variation of the integrand in the sense of Hardy and Krause and the uniformity of sample points measured by star discrepancy. As illustrated in Figure 5, QMC samples are more 'uniform' than MC samples. The randomized quasi-Monte Carlo (RQMC) method, including random shift [27] and scrambling [34], combines the benefits of randomization and low discrepancy sequences to potentially obtain a better result. If the integrand is bounded and smooth, the nested scrambled method achieves a faster convergence rate $O(n^{-3/2+\varepsilon})$ (see [36] for more details). Recently, Ouyang et al. [33] proved that this high convergence rate maintains for unbounded and smooth integrands by using appropriate importance sampling methods. We refer the readers to see more details about QMC in Appendix A.2.

2.2 Reinforcement learning

Reinforcement learning (RL) in continuous state and action space is formulated by a Markov Decision Process (MDP) defined by the tuple $(S, A, p, \rho_0, r, \gamma)$, where S denotes a continuous state space, A denotes a continuous action space, $p(s_{t+1}|s_t, a_t)$ represents the probability of transitioning from state s_t to state s_{t+1} after taking action a_t, ρ_0 is the initial state distribution for s_0, r is a reward function that $r(s_t, a_t)$ returns the reward received after transitioning from state s_t with action a_t, and γ is a discount factor. The goal of maximum entropy reinforcement learning [13, 14, 52] is to learn a policy function $\boldsymbol{\pi} : S \to A$ that maximizes the expected cumulative reward as well as expected entropy of policy,

$$J_{\boldsymbol{\pi}} = \sum_{t=0}^{\infty} \mathbb{E}_{a_t \sim \boldsymbol{\pi}(\cdot|s_t), s_t \sim p_t(\cdot)} \left[r(s_t, a_t) - \alpha \log(\boldsymbol{\pi}(a_t \mid s_t)) \right],$$

where α denotes the relative importance of policy entropy, and p_t is the density function of s_t.

The soft Q function [13], denoted as $Q^{\boldsymbol{\pi}}(s, a)$, is defined as the expected sum of future rewards and entropy, starting from state s, taking action a, and following policy $\boldsymbol{\pi}$ after that. Mathematically, it is expressed as

$$Q^{\boldsymbol{\pi}}(s, a) = r(s, a) + \sum_{t=1}^{\infty} \mathbb{E}_{a_t \sim \boldsymbol{\pi}(\cdot|s_t), s_t \sim p_t(\cdot)} \left[\gamma^t (r(s_t, a_t) - \alpha \log \boldsymbol{\pi}(a_t|s_t)) \mid s_0 = s, a_0 = a \right].$$

In the context of deep reinforcement learning, the $Q^{\boldsymbol{\pi}}(s, a)$ and $\boldsymbol{\pi}(a|s)$ are often represented by neural networks $Q_\theta(s, a)$ and $\boldsymbol{\pi}_\phi(a|s)$, where θ and ϕ are the parameters of neural networks, respectively. The update of the $Q_\theta(s, a)$ is by Temporal Difference learning (TD) [44] as follows,

$$\begin{cases} Q^{\textit{target}} = r + \gamma \mathbb{E}_{a' \sim \boldsymbol{\pi}(\cdot|s')} [Q_\theta(s', a') - \alpha \log \boldsymbol{\pi}(a'|s')], \\ \mathcal{L}(\theta) = \dfrac{1}{2} \mathbb{E}_{\{s,a,r,s'\} \sim \mathcal{D}} \left[\left(Q_\theta(s, a) - Q^{\textit{target}} \right)^2 \right], \\ \theta \leftarrow \theta - l_\theta \nabla_\theta \mathcal{L}(\theta), \end{cases} \tag{1}$$

where $\{s, a, r, s'\}$ is the transition tuple from replay buffer $\mathcal{D}$ and l_θ is the learning rate. The update of policy parameter ϕ is done by gradient ascend over expected cumulative reward $\phi \leftarrow \phi + l_\phi \nabla_\phi J_{\boldsymbol{\pi}}(\phi)$, where l_ϕ is the learning rate for policy parameter. In the framework of Soft Actor-Critic [14], $J_{\boldsymbol{\pi}}(\phi)$ is usually expressed as $J_{\boldsymbol{\pi}}(\phi) = \mathbb{E}_{s, a \sim \boldsymbol{\pi}_\phi(\cdot|s)} [Q_\theta(s, a) - \alpha \log \boldsymbol{\pi}(a \mid s)]$. In off-policy RL and offline RL, the policy gradient is computed by

$$\nabla_\phi J_{\boldsymbol{\pi}}(\phi) = \nabla_\phi \mathbb{E}_{s \sim \mathcal{D}, a \sim \boldsymbol{\pi}_\phi(\cdot|s)} \left[Q_\theta(s, a) - \alpha \log \boldsymbol{\pi}_\phi(a \mid s) \right]. \tag{2}$$

As in [14], $\mathbb{E}_{s \sim \mathcal{D}}$ is estimated by the sample mean of a batch of states from the replay buffer $\mathcal{D}$, and for each s in the batch, one sample Monte Carlo is used to estimate $\mathbb{E}_{a \sim \boldsymbol{\pi}_\phi(\cdot|s)}$.

3 Adaptive two-level quasi-Monte Carlo method

In this section, we provide the details of adaptive two-level quasi-Monte Carlo (ATQ) method in the context of Soft Actor-Critic (SAC).

Our ATQ method focuses on the policy iteration part (2). Without loss of generality, we consider the one-dimensional case. Suppose that $\boldsymbol{\pi}_\phi(\cdot|s_t)$ is the density of a Gaussian distribution with mean

Algorithm 1 ATQ-based SAC

Initialize critic networks $Q_{\theta_1}(s,a)$, $Q_{\theta_2}(s,a)$, actor network $\boldsymbol{\pi}_\phi(s)$ and replay buffer $\mathcal{D}$
Set learning rates $l_\theta, l_\phi, l_\alpha$, set adaptive hyperparameter β, set initial base level parameter b
Set maximum sample size to be 2^M, set batch size N.
for each iteration **do**
  for each environment step **do**
    Sample a_t from $a_t \sim \boldsymbol{\pi}_\phi(\cdot|s_t)$
    Observe s_{t+1}, r_t from environment given a_t
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{D}$
    Record the latest episodic reward $R_{episodic}$
  end for
  for each gradient step **do**
    Sample batch $\{(s^{(j)}, a^{(j)}, r^{(j)}, s'^{(j)})\}_{j=1}^N$ from $\mathcal{D}$
    Update critics θ_1 and θ_2 by (1)
    Adjust the base level parameter b by (11)
    Draw a ξ according to the distribution (10)
    Draw $2^{b+\xi}$ QMC points $\{y_1, \ldots, y_{2^{b+\xi}}\}$
    Compute h by (6) with $\{s^{(j)}\}$ in the batch
    Compute ATQ estimator $G^{\mathrm{QMC}}(b)$ by (9) with the QMC points
    Update actor network $\phi \leftarrow \phi + l_\phi G^{\mathrm{QMC}}(b)$
    Adjust temperature α for entropy regularization
  end for
end for

$\mu_\phi(s_t)$ and variance $(\sigma_\phi(s_t))^2$. We have

$$\log \boldsymbol{\pi}_\phi(x|s_t) = -\log\sqrt{2\pi} - \log\sigma_\phi(s_t) - \frac{(x-\mu_\phi(s_t))^2}{2(\sigma_\phi(s_t))^2}. \tag{3}$$

If $X \sim \mathcal{N}(0,1)$, then $a_t \sim \boldsymbol{\pi}_\phi(\cdot|s_t)$ boils down to

$$a_t = \mu_\phi(s_t) + \sigma_\phi(s_t)X, \tag{4}$$

where μ_ϕ and σ_ϕ are parametric functions, such as neural networks. Under certain mild conditions (usually the integrable condition) that allow us to exchange the order of integration and gradient, the policy gradient of SAC satisfies

$$\begin{aligned}\nabla_\phi J_{\boldsymbol{\pi}}(\phi) &= \nabla_\phi \mathbb{E}_{s_t,X}\left[Q_\theta(s_t, \mu_\phi(s_t)+\sigma_\phi(s_t)X) - \alpha\log\boldsymbol{\pi}\left(\mu_\phi(s_t)+\sigma_\phi(s_t)X|s_t\right)\right] \\ &= \mathbb{E}_{s_t,X}\left[\nabla_\phi Q_\theta(s_t, \mu_\phi(s_t)+\sigma_\phi(s_t)X) - \nabla_\phi\alpha\log\boldsymbol{\pi}\left(\mu_\phi(s_t)+\sigma_\phi(s_t)X|s_t\right)\right] \\ &= \mathbb{E}_X\left[\mathbb{E}_{s_t}\left[\nabla_\phi Q_\theta(s_t, \mu_\phi(s_t)+\sigma_\phi(s_t)X) - \nabla_\phi\alpha\log\boldsymbol{\pi}\left(\mu_\phi(s_t)+\sigma_\phi(s_t)X|s_t\right)\right]\right],\end{aligned} \tag{5}$$

where $s_t \sim \mathcal{D}$, $X \sim \mathcal{N}(0,1)$ and $\log\boldsymbol{\pi}_\phi(\cdot|s_t)$ satisfies (3). Since we can not analytically compute the inter expectation $\mathbb{E}_{s_t}$, we use the sample mean of N samples $\{s_t^{(1)}, \ldots, s_t^{(N)}\}$ form the replay buffer $\mathcal{D}$ to approximate it (this approach is also used in [14]). Denote

$$h(x) = \frac{1}{N}\sum_{j=1}^N \nabla_\phi\left[Q_\theta(s_t^{(j)}, \mu_\phi(s_t^{(j)}) + \sigma_\phi(s_t^{(j)})x) - \alpha\log\boldsymbol{\pi}\left(\mu_\phi(s_t^{(j)}) + \sigma_\phi(s_t^{(j)})x|s_t^{(j)}\right)\right], \tag{6}$$

then an approximation of $\nabla_\phi J_{\boldsymbol{\pi}}(\phi)$ is

$$\widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi) = \mathbb{E}_X\left[h(X)\right]. \tag{7}$$

To compute $\widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi)$, we approximate the expectation $\mathbb{E}_X\left[h(X)\right]$. Differently from the one sample MC approach in [14] (see section 2.2), in the framework of QMC, the following quadrature rule can be used

$$\widehat{I}_n^{\mathrm{QMC}} := \frac{1}{n}\sum_{i=1}^n h\circ\Phi^{-1}(\boldsymbol{y}_i), \tag{8}$$

where $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots\}$ is a low discrepancy sequence (see definitions in Appendix A.2), $\circ$ is the composite operator and Φ is the cumulative distribution function of the standard Gaussian distribution, acting on each component of $\boldsymbol{y}_i$.

Let M satisfy that 2^M is the maximum sample number for hardware limitations and cost tolerance. Our ATQ method inherits the exploration and exploitation concept from SAC, it uses the following two-level estimator to approximate $\mathbb{E}[h(X)]$,

$$G^{\mathrm{QMC}}(b) := \underbrace{\widehat{I}^{\mathrm{QMC}}_{2^b}}_{\text{base level}} + \underbrace{(1/p_\xi)(\widehat{I}^{\mathrm{QMC}}_{2^{b+\xi}} - \widehat{I}^{\mathrm{QMC}}_{2^{b+\xi-1}})}_{\text{stochastic level}}, \tag{9}$$

where in the base level, $b < M$ is a tuning integer, reflecting the extent of exploitation, and in the stochastic level, ξ is a random variable in $\{1, 2, \dots, M-b\}$ with $p_k := \mathbb{P}(\xi = k)$. Moreover, let ξ follow the 'truncated' geometric distribution with

$$p_k = \frac{1}{2^k}, \text{ for } k = 1, \dots, M-b-1, \text{ and } p_{M-b} = \frac{1}{2^{M-b-1}}. \tag{10}$$

The exploratory nature of the stochastic level is manifested in encouraging the estimator to use more samples, making $G^{\mathrm{QMC}}(b)$ an unbiased estimator of $\widehat{I}^{\mathrm{QMC}}_{2^M}$ (see Theorem 1).

The ATQ estimates the policy gradient $\widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi)$ when $Q_\theta(\cdot,\cdot)$ is provided. Since $Q_\theta(\cdot,\cdot)$ is usually not sufficiently accurate at the beginning of the learning process, we need more exploitation at this period. To this end, we proposed an adaptive scheduler to use a large b when the policy is behaving badly (the reward is small) and use a small b when the policy achieves high rewards. More precisely, let b satisfy

$$b = \min\left\{M-1, \left\lfloor (M-1)e^{-\beta R_{episodic}} + 1/2 \right\rfloor\right\}, \tag{11}$$

where $\lfloor x \rfloor$ is the integer part of x, β is a hyperparameter, and $R_{episodic}$ represents the last episodic reward encountered in the training process. This adaptive scheduler of ATQ addresses the intuition that when the policy experiences lower episodic reward, we should perform a better estimation of the gradient, and when the reward is high, indicating that the learning tends to be stable, we use a smaller b to save computational cost. Such an adaptive strategy allows our ATQ method to allocate resources effectively.

4 Convergence analysis of the ATQ method

In this section, we study the convergence of the ATQ method. To illustrate the superiority of the QMC method, we compare $G^{\mathrm{QMC}}(b)$ with the corresponding MC estimator $G^{\mathrm{MC}}(b)$ with the definition

$$G^{\mathrm{MC}}(b) := \widehat{I}^{\mathrm{MC}}_{2^b} + (1/p_\xi)\left(\widehat{I}^{\mathrm{MC}}_{2^{b+\xi}} - \widehat{I}^{\mathrm{MC}}_{2^{b+\xi-1}}\right), \tag{12}$$

where $\widehat{I}^{\mathrm{MC}}_n := \frac{1}{n}\sum_{i=1}^n h(X_i)$ with $\{X_i\}$ being identically and independently distributed samples from the standard Gaussian distribution. We make some assumptions about the functions that contribute h.

Assumption 1. *For every θ and ϕ, $Q_\theta, \mu_\phi, \sigma_\phi$ are smooth and their derivatives are bounded by a polynomial. Moreover, there is a $\delta > 0$ such that σ_ϕ is bounded below by δ.*

The assumption holds if we choose the networks appropriately. For example, let $Q_\theta, \mu_\phi, \sigma_\phi$ be multilayer perceptrons (MLPs) with smooth and bounded activation functions. By the appendix in Ouyang et al. [33], it is easy to verify that they satisfy Assumption 1 under some mild conditions about the parameters in networks. The following theorem presents the convergence rates of $G^{\mathrm{QMC}}(b)$, $G^{\mathrm{MC}}(b)$, $\widehat{I}^{\mathrm{QMC}}_{2^m}$ and $\widehat{I}^{\mathrm{MC}}_{2^m}$. Denote $\mathbb{E}_\xi$ the expectation with respect to ξ.

Theorem 1. *Assume Assumption 1 holds. Suppose that $h(X)$ has finite variance, and ξ is independent of X, satisfying* (10)*. Then for every fixed integer b, we have the following results.*

1. *$G^{\mathrm{QMC}}(b)$ is an unbiased estimator of $\widehat{I}^{\mathrm{QMC}}_{2^M}$ with respect to ξ, i.e., $\mathbb{E}_\xi\left[G^{\mathrm{QMC}}(b)\right] = \widehat{I}^{\mathrm{QMC}}_{2^M}$. Moreover, if we use nested scrambled sobol' sequences in* (8) *and* (9)*, then for any $\varepsilon > 0$,*

$$\sqrt{\mathbb{E}\left[\left|G^{\mathrm{QMC}}(b) - \widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi)\right|^2\right]} = O(2^{-b+\varepsilon}), \tag{13}$$

and for every integer m,

$$\sqrt{\mathbb{E}\left[\left|\widehat{I}_{2^m}^{\mathrm{QMC}} - \widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi)\right|^2\right]} = O(2^{-m+\varepsilon}). \tag{14}$$

2. *$G^{\mathrm{MC}}(b)$ is an unbiased estimator of $\widehat{I}_{2^M}^{\mathrm{MC}}$ with respect to ξ, and thus it is an unbiased estimator of $\widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi)$. Moreover,*

$$\sqrt{\mathbb{E}\left[\left|G^{\mathrm{MC}}(b) - \widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi)\right|^2\right]} = O(2^{-b/2}), \tag{15}$$

and for every integer m,

$$\sqrt{\mathbb{E}\left[\left|\widehat{I}_{2^m}^{\mathrm{MC}} - \widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi)\right|^2\right]} = O(2^{-m/2}). \tag{16}$$

Proof. The proof of this theorem is presented in Appendix B. □

As shown in (9), the ATQ method has the base level and the stochastic level with the purpose of exploitation and exploration, respectively. Theorem 1 indicates the efficiency of exploitation of $G^{\mathrm{QMC}}(b)$, thanks to QMC, which has the convergence rate $O(2^{-b+\varepsilon})$ in the sense of root mean squared error (RMSE). In contrast, the RMSE of MC method $G^{\mathrm{MC}}(b)$ is only $O(2^{-b/2})$. Therefore, we choose QMC method in our ATQ in order to obtain a more efficient exploitation (a better convergence rate).

Note that by setting $m = M$ in (14), $\widehat{I}_{2^M}^{\mathrm{QMC}}$ is the most accurate estimation of the policy gradient $\widehat{\nabla}_\phi J_{\boldsymbol{\pi}}(\phi)$. However, this optimal estimator is costly. Luckily, due to the effect of the stochastic level, G^{QMC} is an unbiased estimator of $\widehat{I}_{2^M}^{\mathrm{QMC}}$. Therefore, our ATQ method explores the efficiency of $\widehat{I}_{2^M}^{\mathrm{QMC}}$ and will have better results in the average sense.

Moreover, in the mean sense, the sample size n for ATQ is

$$n = \sum_{i=1}^{M-b-1} 2^{b+i} \cdot 2^{-i} + 2^{b+M-b} \cdot 2^{-M+b+1} = (M-b-1)2^b + 2^{b+1} = (M-b+1)2^b.$$

From this perspective, our ATQ method is efficient, given that for fixed $M > 0$, it only uses $O(2^b)$ samples, to achieve the RMSE rate $O(2^{-b+\varepsilon})$, which is better than MC methods. If we dynamically select b by (11) during the learning process, then the ATQ method uses significantly fewer samples in aggregate. This observation is validated by our experimental results in Section 5 (see Figure 4).

5 Experiments

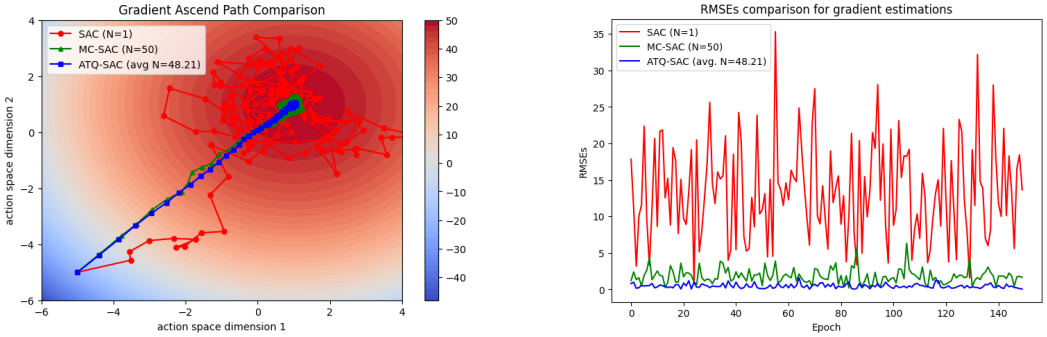
We start with a toy experiment to show how ATQ aids convergence. And then, we consider online Mujoco environments and offline D4RL datasets to compare the performance of ATQ-based SAC with state-of-the-art algorithms. Moreover, a detailed ablation study about ATQ and adaptive two-level MC, plain QMC [2], one stochastic level QMC is presented in the subsection 5.3. In the last subsection, we study the performance of ATQ-based SAC under insufficient data. The implementation details of the proposed ATQ method are discussed in Appendix C.2. The source code will be made public upon acceptance.

5.1 Toy Experiment

We introduce a toy RL problem to demonstrate how ATQ accelerates convergence. In this simplified setting, we assume that $\tilde{Q}(s,a) = Q(s,a) - \alpha \log \boldsymbol{\pi}(a|s)$ is a known function, s is fixed, and $a_t = \phi + \sigma X$ where $X \sim \mathcal{N}(0, \mathbf{I}_2)$. It can be viewed as setting $\mu_\phi(s_t) = \phi$ and $\sigma_\phi(s_t) = \sigma$ in Equation (4). In this specific case, we set $\tilde{Q}(s, \boldsymbol{a}) = -\left\|\boldsymbol{a} - \mathbf{1}\right\|^2 + 50$, $\sigma = 5$. Then we have

$\nabla_\phi J_{\boldsymbol{\pi}}(\phi) = \mathbb{E}_X\left[\nabla_\phi \tilde{Q}(s, \phi + \sigma X)\right]$. Reinforcement learning can then be simplified to perform gradient ascend over $J_{\boldsymbol{\pi}}(\phi)$. From this setting, we can derive that the optimal policy parameter is $\phi = (1, 1)$. Thus the convergence of RL is then simplified to the convergence of ϕ to the static point $(1, 1)$ in the policy parameter space.

In our toy experiment, we compare the performance of SAC, MC-based SAC, and our proposed ATQ-based SAC, which use $\widehat{I}_1^{\mathrm{MC}}$, $\widehat{I}_{50}^{\mathrm{MC}}$ and $G^{\mathrm{QMC}}(4)$ to estimate the policy gradient $\nabla_\phi J_{\boldsymbol{\pi}}(\phi)$, respectively. Figure 1a visualizes the gradient ascend steps. The contour map illustrates the value of $\tilde{Q}(s, a)$ across the action space. Additionally, since $a_t = \phi + \sigma X$, we depict the update of ϕ on the same map. From the result in Figure 1a, it is clear that ATQ-based SAC achieves the most stable convergence. Figure 1b demonstrates the RMSEs curve of $\widehat{I}_1^{\mathrm{MC}}$, $\widehat{I}_{50}^{\mathrm{MC}}$ and $G^{\mathrm{QMC}}(4)$ in the learning period. Our ATQ-based SAC uses fewer samples (average 48.21) and achieves significantly better convergence results in the sense of RMSE. This experiment validates the theoretical result in Theorem 1.

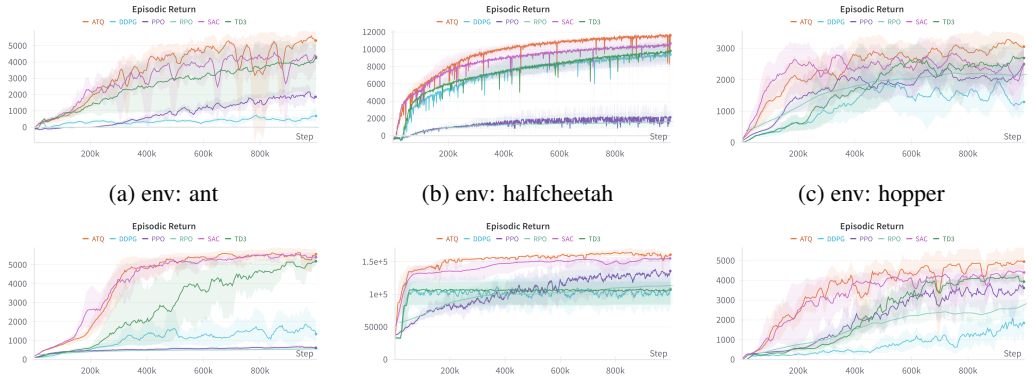


(a) Gradient ascend path comparison (b) RMSEs comparison for gradient estimations

Figure 1: Comparison of gradient ascend path and RMSEs

5.2 Experiment result for Online and Offline RL

In this subsection, we elaborate on our experiment settings and results for online RL environments and offline RL datasets. The implementation of our proposed ATQ method remains the same for online and offline RL, while the backbone SAC network slightly differs in the two settings. For the online setting, the backbone SAC is a standard SAC with two critic networks [14]. For the offline setting, the backbone SAC is SAC-N which increases the number of critic networks to N [1]. Detailed implementation of ATQ-based SAC is in Appendix C.2.



(a) env: ant (b) env: halfcheetah (c) env: hopper

(d) env: humanoid (e) env: humanoidstandup (f) env: walker2d

Figure 2: Training return curve for online algorithms.

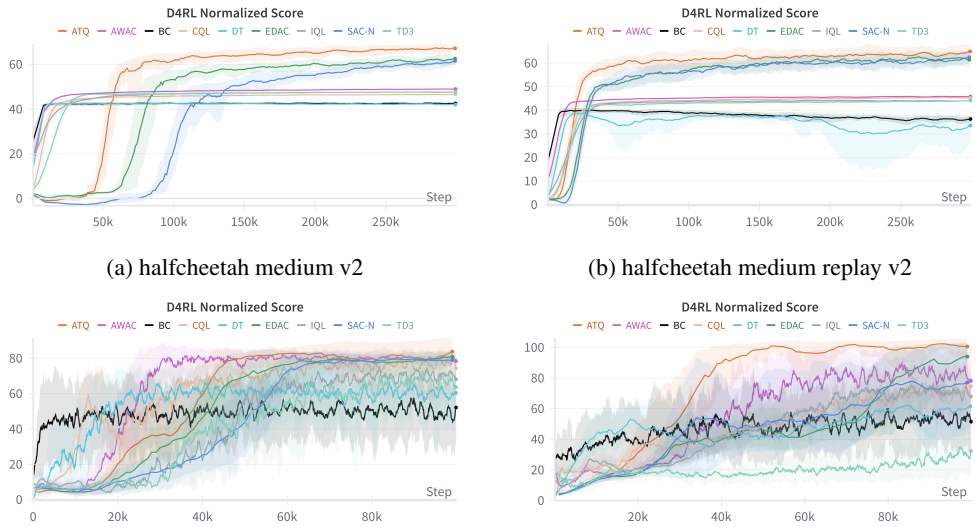
Environment and datasets. In this section, the performance of the proposed ATQ-based SAC is evaluated on the suite of the classic Mujoco [46] environment and D4RL Mujoco datasets[8]. For the offline experiment, to better demonstrate the behavior of ATQ method, we focus on suboptimal datasets such as medium-replay and medium. This is because a relatively smaller number of critics networks is required for the backbone SAC-N network [1].

Online RL Baselines. In the online setting, we compare the performance of our proposed algorithm with Proximal Policy Gradient (PPO) [41], Deep Deterministic Policy Gradient (DDPG) [28], Robust Policy Optimization (RPO) [38], TD3 [10], and Soft Actor-Critic (SAC) [14]. Appendix C.1 provides a more detailed introduction about the baselines. A more detailed comparison using QMC samples with fixed sample numbers in gradient estimation [2] will be presented in the ablation study section.

Offline RL Baselines. In the offline setting, we consider offline algorithms including Behavior Cloning (BC), Decision Transformer (DT) [4], SAC-N [1], EDAC, TD3+BC [9], Implicit Q Learning (IQL) [23], Conservative Q Learning (CQL) [24], and AWAC [30]. A detailed discussion of the offline baseline can be found in Appendix C.1.

Hardwares. All the experiments are run on regular computer resources such as NVIDIA RTX 3090 GPUs. The regular training run time for our proposed method is around 5 hours. More detailed information can be found in the Appendix C.1.

Online RL Results. Initially, we present the training curves from our online experiments. The algorithms were trained using three different random seeds. The solid line in the figures represents the average episodic return across these seeds during training, while the shaded region indicates the range between the minimum and maximum returns. As illustrated in Figure 2, ATQ-based SAC outperforms other baseline algorithms across all six environments. Specifically, ATQ-based SAC demonstrates greater stability in the training process, as evidenced by lower variance in environments such as Halfcheetah and Humanoidstandup.



(a) halfcheetah medium v2

(b) halfcheetah medium replay v2

(c) walker medium replay v2

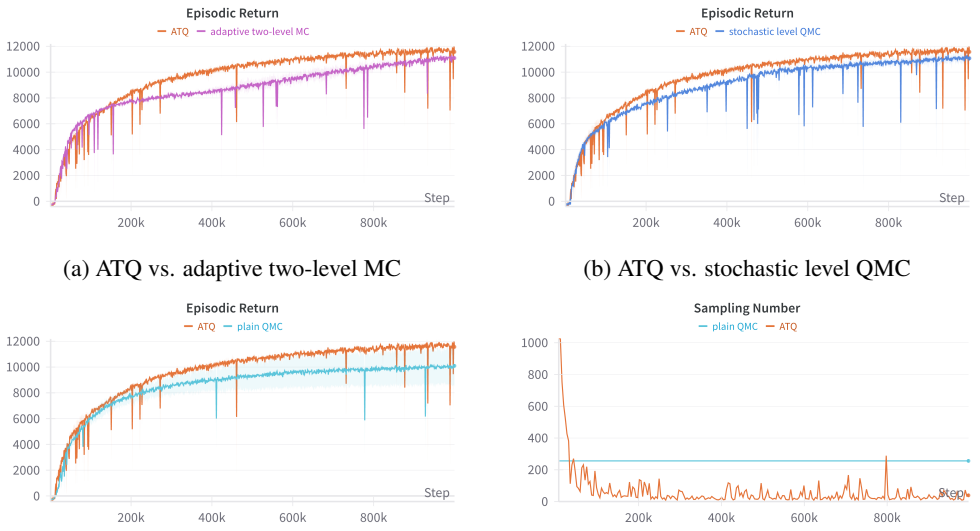
(d) hopper medium replay v2

Figure 3: Evaluation normalized d4rl score curve for selected offline algorithms.

Offline RL Results. Additionally, we report the evaluation score curve for the offline dataset recorded during training. The evaluation score, represented as the D4RL normalized score [8], is measured during evaluation rollouts at specified intervals throughout the training process. Each algorithm is run with 4 different random seeds. As depicted in Figure 3, ATQ demonstrates consistency with the findings from the prior online experiments, showing that it provides faster convergence.

5.3 Ablation Study

In this subsection, we set the maximum sample size to be 2^{10}, i.e., $M = 10$ in (9), (10) and (11).



(a) ATQ vs. adaptive two-level MC

(b) ATQ vs. stochastic level QMC

(c) ATQ vs. plain QMC: return

(d) ATQ vs. plain QMC: sample sizes

Figure 4: Ablation Study Figures.

ATQ vs. adaptive two-level MC. We discuss the role of the QMC method within ATQ. For this purpose, we substitute QMC points with MC points to obtain the adaptive two-level MC method. In other words, we are comparing two estimators, $G^{\mathrm{QMC}}(b)$ and $G^{\mathrm{MC}}(b)$ (see definition (12)) with b satisfying (11). In Theorem 1, we prove that $G^{\mathrm{QMC}}(b)$ has faster convergence rates than $G^{\mathrm{MC}}(b)$. We validate the theoretical advantages of ATQ through experimental verification in the HalfCheetah environment. From Figure 4a we can see that ATQ outperforms the adaptive two-level MC method.

ATQ vs. plain QMC. We delve into the role of the two-level structure within ATQ. We compare the training curve of ATQ-based SAC to QMC-based SAC [2] in the HalfCheetah environment. In our experiment setting, the base level parameter b follows (11) for ATQ method $G^{\mathrm{QMC}}(b)$, and the QMC-based SAC uses the plain QMC estimator $\widehat{I}_{2^8}^{\mathrm{QMC}}$ with the sample size 256. The dynamic sample sizes curve is displayed in Figure 4d. The curve shows that ATQ utilizes more samples in the early stages of training and gradually reduces the number of samples thereafter, thus achieving a reduction in computational costs. The average sample size of ATQ-based SAC is 65.68, which is sufficiently lower than QMC-based SAC [2] (which is 256). Figure 4c shows that ATQ-based SAC achieves better performance than QMC-based SAC [2], while uses less samples in aggregate.

ATQ vs. stochastic level QMC. We now study the effect of the base level in ATQ. We compare the performance differences between ATQ and stochastic level QMC (lack of the base level). The role of the base level is to explore; as can be seen from Theorem 1, the base level directly impacts the convergence rate. From the Figure 4b, it is apparent that without the base level, stochastic level QMC performs worse than ATQ.

5.4 The performance of ATQ-based SAC under insufficient data

Insufficient data is a common challenge, particularly when interaction with the environment is slow and expensive in an online setting, or when the dataset size is limited in an offline setting. In this experiment, we used various percentages of D4RL halfcheetah-medium suboptimal data to train the baseline methods and our ATQ method. We reported the average normalized score over 100 evaluation rollouts after the training was completed. According to Table 1, ATQ demonstrates robust performance even when only 10% of the dataset is used, outperforming other baseline algorithms.

6 Conclusion and future work

In conclusion, we propose an adaptive two-level quasi-Monte Carlo method to approximate policy gradient. The ATQ method inherits the concept of exploitation and exploration from SAC. It uses one

Table 1: Evaluation Score under Different Levels of Data Insufficiency

Used Data Percentage	**10%**	**30%**	**50%**
ATQ (Ours)	**58.32** $\pm$ 0.52	**61.52** $\pm$ 0.31	**65.20** $\pm$ 4.89
SAC-N	48.06 $\pm$ 0.13	53.78 $\pm$ 0.13	56.46 $\pm$ 0.55
IQL	46.49 $\pm$ 0.94	47.00 $\pm$ 0.70	47.13 $\pm$ 0.86
CQL	45.93 $\pm$ 0.75	46.30 $\pm$ 0.73	46.03 $\pm$ 0.76
BC	36.32 $\pm$ 9.08	40.89 $\pm$ 6.36	42.04 $\pm$ 1.34

base level to ensure the high convergence rate and one stochastic level to make ATQ an unbiased estimator of the optimal one $\widehat{I}_{2^M}^{\mathrm{QMC}}$. Experimentally, the ATQ-based SAC outperforms other strong baselines in online and offline reinforcement learning tasks. A direct line of future work is to refine the adaptive mechanism and study the multi-level quasi-Monte Carlo method in SAC framework. Other future directions may related to exploring quasi-Monte Carlo methods in multi-agent RL.

References

[1] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

[2] Sébastien MR Arnold, Pierre L'Ecuyer, Liyu Chen, Yi-fan Chen, and Fei Sha. Policy learning and evaluation with randomized quasi-Monte Carlo. *arXiv preprint arXiv:2202.07808*, 2022.

[3] Russel E. Caflisch, William J. Morokoff, and Art B. Owen. Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. *Journal of Computational Finance*, 1:27–46, 1997.

[4] 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, 2021.

[5] Josef Dick, Christian Irrgeher, Gunther Leobacher, and Friedrich Pillichshammer. On the optimal order of integration in Hermite spaces with finite smoothness. *SIAM Journal on Numerical Analysis*, 56(2):684–707, 2018.

[6] Josef Dick and Friedrich Pillichshammer. *Digital Nets and Sequences. Discrepancy Theory and Quasi–Monte Carlo Integration*. Cambridge University Press, Cambridge, 2010.

[7] Josef Dick, Daniel Rudolf, and Houying Zhu. A weighted discrepancy bound of quasi-Monte Carlo importance sampling. *Statistics & Probability Letters*, 149:100–106, 2019.

[8] 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.

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

[10] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[11] Michael B. Giles. Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328, 2015.

[12] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, New York, 2004.

[13] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[15] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[16] Chang han Rhee and Peter W. Glynn. Unbiased estimation with square root convergence for SDE models. *Oper. Res.*, 63:1026–1043, 2015.

[17] Zhijian He, Hejin Wang, and Xiaoqun Wang. Quasi-Monte Carlo and importance sampling methods for bayesian inverse problems. *arXiv preprint arXiv:2403.11374*, 2024.

[18] Zhijian He and Xiaoqun Wang. Good path generation methods in quasi-Monte Carlo for pricing financial derivatives. *SIAM Journal on Scientific Computing*, 36(2):B171–B197, 2014.

[19] Edmund Hlawka and R Mück. Über eine transformation von gleichverteilten folgen II. *Computing*, 9:127–138, 1972.

[20] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

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

[24] 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.

[25] Frances Y Kuo, Ian H Sloan, Grzegorz W Wasilkowski, and Benjamin J Waterhouse. Randomly shifted lattice rules with the optimal rate of convergence for unbounded integrands. *Journal of Complexity*, 26(2):135–160, 2010.

[26] Frances Y Kuo, Ian H Sloan, and Henryk Woźniakowski. Lattice rule algorithms for multivariate approximation in the average case setting. *Journal of Complexity*, 24(2):283–323, 2008.

[27] Frances Y Kuo, Grzegorz W Wasilkowski, and Benjamin J Waterhouse. Randomly shifted lattice rules for unbounded integrands. *Journal of Complexity*, 22(5):630–651, 2006.

[28] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[29] Yang Liu and Raúl Tempone. Nonasymptotic convergence rate of quasi-Monte Carlo: Applications to linear elliptic PDEs with lognormal coefficients and importance samplings. *arXiv preprint arXiv:2310.14351*, 2023.

[30] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[31] James A Nichols and Frances Y Kuo. Fast CBC construction of randomly shifted lattice rules achieving $\mathcal{O}(n^{-1+\delta})$ convergence for unbounded integrands over $\mathbb{R}^s$ in weighted spaces with POD weights. *Journal of Complexity*, 30(4):444–468, 2014.

[32] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, PA, 1992.

[33] Du Ouyang, Xiaoqun Wang, and Zhijian He. Quasi-Monte Carlo for unbounded integrands with importance sampling. *arXiv preprint arXiv:2310.00650*, 2023.

[34] Art B Owen. Scrambled net variance for integrals of smooth functions. *The Annals of Statistics*, 25(4):1541–1562, 1997.

[35] Art B Owen. Halton sequences avoid the origin. *SIAM Review*, 48(3):487–503, 2006.

[36] Art B. Owen. Local antithetic sampling with scrambled nets. *The Annals of Statistics*, 36(5):2319–2343, 2008.

[37] Art B Owen. *Monte Carlo Theory, Methods and Examples*. Stanford, 2013.

[38] Md Masudur Rahman and Yexiang Xue. Robust policy optimization in deep reinforcement learning. *arXiv preprint arXiv:2212.07536*, 2022.

[39] Chang-han Rhee and Peter W Glynn. A new approach to unbiased estimation for sde's. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pages 1–7. IEEE, 2012.

[40] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[41] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[42] 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.

[43] Wesley A Suttle, Amrit Bedi, Bhrij Patel, Brian M Sadler, Alec Koppel, and Dinesh Manocha. Beyond exponentially fast mixing in average-reward reinforcement learning via multi-level Monte Carlo actor-critic. In *International Conference on Machine Learning*, pages 33240–33267. PMLR, 2023.

[44] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.

[45] Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. CORL: Research-oriented deep offline reinforcement learning library. In *3rd Offline RL Workshop: Offline RL as a "Launchpad"*, 2022.

[46] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[47] Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023.

[48] Hejin Wang and Zhan Zheng. Randomly shifted lattice rules with importance sampling and applications. *Mathematics*, 12(5):630, 2024.

[49] Jichang Xiao, Fengjiang Fu, and Xiaoqun Wang. Analysis of the generalization error of deep learning based on randomized quasi-Monte Carlo for solving linear kolmogorov PDEs. *arXiv preprint arXiv:2310.18100*, 2023.

[50] Ye Xiao and Xiaoqun Wang. Conditional quasi-Monte Carlo methods and dimension reduction for option pricing and hedging with discontinuous functions. *Journal of Computational and Applied Mathematics*, 343:289–308, 2018.

[51] Shangtong Zhang, Romain Laroche, Harm van Seijen, Shimon Whiteson, and Remi Tachet des Combes. A deeper look at discounting mismatch in actor-critic algorithms. *arXiv preprint arXiv:2010.01069*, 2020.

[52] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

A Supplementary introduction

A.1 Related work

Reinforcement learning. The Deep Deterministic Policy Gradient (DDPG) algorithm [28] based on this architecture learns policies in high-dimensional, continuous action spaces and it is an extension of the earlier Deterministic Policy Gradient (DPG) algorithms [42], combining ideas from DPG and deep Q-networks. To overcome the problems of Trust Region Policy Optimization (TRPO) [40], Schulman et al. [41] proposed Proximal Policy Optimization (PPO) method. PPO is an on-policy algorithm that improves upon the stability and simplicity of policy gradient methods, providing an easier-to-tune but powerful method for training deep reinforcement learning policies. Inspired by the DDPG algorithm, Fujimoto et al. [10] proposed the Twin Delayed Deep Deterministic policy gradient algorithm (TD3). TD3 improves upon DDPG by addressing the function approximation errors through the use of twin Q-networks and delayed policy updates, enhancing learning stability.

Soft Actor-Critic. Regarding policy regularization, Haarnoja et al. [14] proposed an entropy regularization term in their seminal work on Soft Actor–Critics (SACs) which is still a state-of-the-art algorithm in model-free, off-policy reinforcement learning optimized for environments with continuous action spaces. Haarnoja et al. [15] introduced an extension to the original SAC algorithm that includes an automatic mechanism to adjust the temperature parameter. This modification aims to automate the tuning of the entropy coefficient, making SAC more adaptive and easier to deploy across different tasks without manual tuning of hyperparameters. Zhang et al. [51] improved the explanation of SAC at the theoretical level and this research provided deeper theoretical insights into actor-critic methods, including SAC, focusing on issues like discounting mismatch and how it affects the convergence and performance of these algorithms.

A.2 Quasi-Monte Carlo

Many problems in statistics, financial engineering, machine learning, and reinforcement learning involve calculating expectations. How to numerically solve integration problems efficiently is key to improving computational efficiency. For numerically computing the integral $\mathbb{E}[f(Z)]$, where $Z \sim \mathcal{U}[0,1]^d$, a commonly used estimator is

$$\widehat{I}_n := \frac{1}{n}\sum_{i=1}^{n} f(y_i).$$

The choice of different quadrature points corresponds to different methods. Monte Carlo (MC) method use random points, i.e., $\{y_i\}_{i=1}^n$ are independent and identically distributed random samples from $\mathcal{U}[0,1]^d$. In this situation, $\widehat{I}_n$ is an unbiased estimator of $\mathbb{E}[f(Z)]$. Moreover, if $f(Z)$ has finite variance, then MC method achieves the convergence rate $O(n^{-1/2})$ due to the central limit theorem. Unlike the MC method, the quasi-Monte Carlo (QMC) method uses low-discrepancy point sets. The error rate of QMC is based on the following Koksma-Hlawka inequality [19]

$$\left|\int_{[0,1]^d} f(y)\mathrm{d}y - \frac{1}{n}\sum_{i=1}^{n} f(y_i)\right| \le V_{\mathrm{HK}}(f) D_n^*\left(\{y_1,\ldots,y_n\}\right), \tag{17}$$

where $V_{\mathrm{HK}}(f)$ is the variation of f in the sense of Hardy and Krause and $D_n^*\left(\{y_1,\ldots,y_n\}\right)$ is the star discrepancy of the point set $\{y_1,\ldots,y_n\}$. In order to provide a complete explanation of (17), we need the following definitions.

Define $1:d$ to be the set $\{1,2,\ldots,d\}$. For a subset $\boldsymbol{u} \subseteq 1:d$, let $|\boldsymbol{u}|$ denote the cardinality of $\boldsymbol{u}$. For $\boldsymbol{a} = (a_1,\ldots,a_d)$ and $\boldsymbol{b} = (b_1,\ldots,b_d)$, the vector $\boldsymbol{a}^{\boldsymbol{u}} : \boldsymbol{b}^{-\boldsymbol{u}}$ is then defined such that its j-th component is a_j if $j \in \boldsymbol{u}$, and b_j otherwise. Denote mixed derivatives as $\partial^{\boldsymbol{u}} := \prod_{i\in\boldsymbol{u}} \partial/\partial x_i$. We call f a smooth function if for every $\boldsymbol{u} \in 1:d$, $\partial^{\boldsymbol{u}} f$ is continuous.

If f is smooth, then $V_{\mathrm{HK}}(f)$ is defined by (see [32] for a general definition)

$$V_{\mathrm{HK}}(f) := \sum_{\varnothing \neq \boldsymbol{u} \in 1:d} \int_{[0,1]^d} \left| \partial^{\boldsymbol{u}} f(\boldsymbol{y}^{\boldsymbol{u}} : \mathbf{1}^{-\boldsymbol{u}}) \right| \mathrm{d}\boldsymbol{y},$$

where $\mathbf{1} := (1, \ldots, 1)$ is the vector of d ones and $\boldsymbol{y} = (y_1, \ldots, y_d)$.

The key to the QMC method lies in the construction of low discrepancy sequences. There are several kinds of low discrepancy sequences, such as Faure sequence, Sobol' sequence and Halton sequence with their first n points achieving star discrepancy of $O(n^{-1}(\log n)^d)$ (see [12, 32, 37] for more details). The star discrepancy measures the uniformity of sample points. In Figure 5, the left side shows random samples, while the right side displays a Sobol' sequence. It can be observed that the specially constructed Sobol' sequence is more 'uniform' than the random sequence.

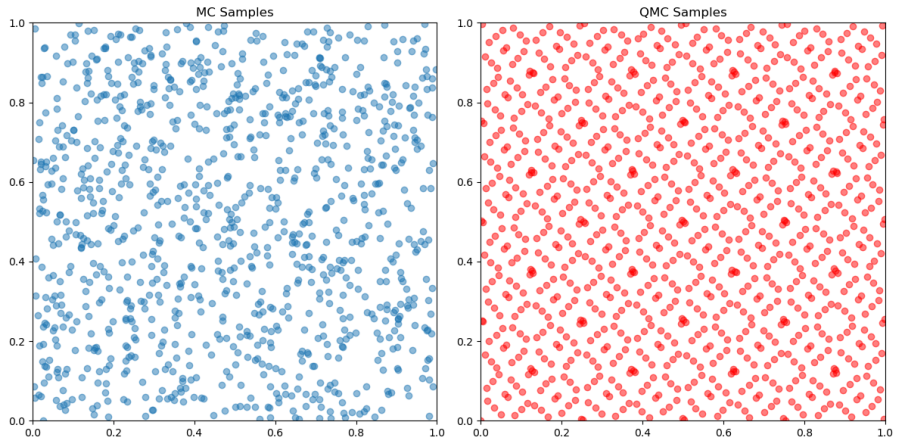


Figure 5: MC samples and QMC samples

As a result, if $V_{\mathrm{HK}}(f) < \infty$, then it follows from (17) that the QMC methods achieve a convergence rate of $O(n^{-1+\varepsilon})$ with ε being arbitrarily small, while the MC convergence rate is $O(n^{-1/2})$. Theoretical results indicate that QMC converges faster than MC.

The randomize quasi-Monte Carlo (RQMC) method is a technique that randomizes QMC point sequences while maintaining the low discrepancy. Common RQMC methods include scrambling and random shifts (see [32, 35, 37, 27, 25]). In this paper, we consider the nested scrambled Sobol' point set $\{y_1, \ldots, y_n\}$ (see [34, 36, 37]), which satisfies

1. $\forall 1 \leq i \leq n$, $y_i \sim \mathcal{U}[0,1]^d$;
2. there is a constant C independent of n such that

$$D_n^* \left(\{y_1, \ldots, y_n\} \right) \leq C \frac{(\log n)^{d-1}}{n}, \quad a.s. \tag{18}$$

Every point of RQMC point sets is uniformly distributed on $[0,1]^d$, but they are not independent. They are correlated in order to keep the low discrepancy. By (18), RQMC point set is also a low discrepancy point set, thus, RQMC methods also have the convergence rate $O(n^{-1+\varepsilon})$. Moverover, if the integrand is smooth and bounded, Owen [36] proved the nested scrambled RQMC method achieves the convergence rate of $O(n^{-3/2+\varepsilon})$. Ouyang et al. [33] proved the $O(n^{-3/2+\varepsilon})$ convergence rate also holds for smooth and unbounded integrands if applying appropriate importance sampling methods.

Additional work on QMC includes the study of lattice rule [5, 6, 7, 17, 25, 26, 27, 31, 48], investigation of the non-asymptotic convergence rates of QMC methods [29], the study of the effects of Brownian path generation and dimension reduction methods on QMC [18, 50], and the efficiency of QMC methods for solving partial differential equations by deep learning [49].

B Proof

Proof of Theorem 1. For the proof of the QMC, note that

$$\begin{aligned}\mathbb{E}_\xi\left[G^{\mathrm{QMC}}(b)\right] &= \mathbb{E}_\xi\left[\widehat{I}_b^{\mathrm{QMC}} + \frac{1}{p_\xi}\left(\widehat{I}_{2^{b+\xi}}^{\mathrm{QMC}} - \widehat{I}_{2^{b+\xi-1}}^{\mathrm{QMC}}\right)\right] \\ &= \widehat{I}_{2^b}^{\mathrm{QMC}} + \sum_{i=1}^{M-b} p_i \frac{1}{p_i}\left(\widehat{I}_{2^{b+i}}^{\mathrm{QMC}} - \widehat{I}_{2^{b+i-1}}^{\mathrm{QMC}}\right) = \widehat{I}_{2^M}^{\mathrm{QMC}}.\end{aligned}$$

Therefore, $G^{\mathrm{QMC}}(b)$ is the unbiased estimator of $\widehat{I}_{2^M}^{\mathrm{QMC}}$. For the proof of (14), under Assumption 1, it is easy to verify that $h(x)$ and its derivatives are bounded by a polynomial. In fact, for $\boldsymbol{u} \subset 1:d$ (see definitions in Appendix A.2),

$$\begin{aligned}|\partial^{\boldsymbol{u}} h(x)| \le \frac{1}{N}\sum_{j=1}^{N}\Bigg(&\left|\partial^{\boldsymbol{u}}\nabla_\phi Q_\theta(s_t^{(j)}, \mu_\phi(s_t^{(j)}) + \sigma_\phi(s_t^{(j)})x)\right|\left|\sigma_\phi(s_t^{(j)})\right| \\ &+ \alpha\partial^{\boldsymbol{u}}\nabla_\phi\left|\log\boldsymbol{\pi}\left(\mu_\phi(s_t^{(j)}) + \sigma_\phi(s_t^{(j)})x\middle|s_t^{(j)}\right)\right|\Bigg).\end{aligned} \tag{19}$$

Due to $\log\boldsymbol{\pi}(x|s_t)$ is a quadratic function with respect to x (see (3)) and the derivatives of Q_θ are bounded by a polynomial, the right hand side of (19) can be bounded by $A|x|^k + B$ for some constant $A, B > 0$ and integer $k \ge 1$ that independent of $\boldsymbol{u}$. Noting that the nested scrambled Sobol' sequence satisfies (18), by the corollary 4.10 of Ouyang et al. [33], we prove the desired result.

Based on (14), we next prove (13). Notice that

$$\mathbb{E}\left[\left|G^{\mathrm{QMC}} - \widehat{\nabla}J_{\boldsymbol{\pi}}(\phi)\right|^2\right] = \mathbb{E}\left[\mathbb{E}\left[\left|\widehat{I}_{2^b}^{\mathrm{QMC}} + \frac{1}{p_\xi}\left(\widehat{I}_{2^{b+\xi}}^{\mathrm{MC}} - \widehat{I}_{2^{b+\xi-1}}^{\mathrm{QMC}}\right) - \nabla J_{\boldsymbol{\pi}}(\phi)\right|^2\middle|\xi\right]\right], \tag{20}$$

and

$$\begin{aligned}&\mathbb{E}\left[\left|\widehat{I}_{2^b}^{\mathrm{QMC}} + \frac{1}{p_\xi}\left(\widehat{I}_{2^{b+\xi}}^{\mathrm{QMC}} - \widehat{I}_{2^{b+\xi-1}}^{\mathrm{QMC}}\right) - \nabla J_{\boldsymbol{\pi}}(\phi)\right|^2\middle|\xi\right] \\ &\le 3\mathbb{E}\left[\left|\widehat{I}_{2^b}^{\mathrm{QMC}} - \widehat{\nabla}J_{\boldsymbol{\pi}}(\phi)\right|^2\right] + 3\frac{1}{p_\xi}\left(\mathbb{E}\left[\left|\widehat{I}_{2^{b+\xi}}^{\mathrm{QMC}} - \widehat{\nabla}J_{\boldsymbol{\pi}}(\phi)\right|^2\middle|\xi\right] + \mathbb{E}\left[\left|\widehat{I}_{2^{b+\xi-1}}^{\mathrm{QMC}} - \widehat{\nabla}J_{\boldsymbol{\pi}}(\phi)\right|^2\middle|\xi\right]\right) \\ &\le C\left(2^{-2b} + \frac{1}{p_\xi}\left(2^{-2(b+\xi)} + 2^{-2(b+\xi-1)}\right)\right),\end{aligned}$$

where C is a constant. It follows from (20) that

$$\mathbb{E}\left[\left|G^{\mathrm{MC}} - \widehat{\nabla}J_{\boldsymbol{\pi}}(\phi)\right|^2\right] \le C2^{-2b}\left(1 + \frac{5}{4}\sum_{k=1}^{m}\left(2^{-k}\right)\right) = O(2^{-2b}). \tag{21}$$

For the QMC part, we can apply the same method as in the proof of (15). Noting that $h(X)$ has finite variance, the proof of (15) and (16) is straightforward. □

C Experiment Details

C.1 Experiment Setups

Online Experiments Setup. For the online experiments, we utilized the default Mujoco implementation provided by the Gymnasium project [47]. Although all environments were tested, our work reports results from only six environments. This selection was made because the difficulty level of the other environments was too low, resulting in uniformly high performance across all algorithms, making it difficult to distinguish the performance of individual algorithms.

Offline Experiment Setup. For the offline experiments, we take the D4RL benchmark[8]. In our work, we focus on suboptimal offline datasets. The selection of datasets is because the backbone network we used in offline experiments requires a large number (up to 500) of critic networks[1] for

other datasets. Due to computation resource limit, we focus our work on suboptimal offline datasets where relatively less critic networks are needed.

Baselines. For the online RL baselines, a collection of high-performing algorithms in continuous action space is selected. The implementation of online baselines is based on an open-sourced baseline project [20]. For the offline RL baselines, The implementation of offline baselines is based on the open-source baseline library[45].

To ensure fairness among all compared online baselines, we have the following constraints. Firstly, the batch size for off-policy algorithms such as ATQ-based SAC, SAC, TD3, and DDPG is set to 256. Secondly, the policy learning rate is set to 0.0003 for all the online algorithms. Thirdly, all the algorithms use the same Adam optimizer [21].

Similarly, to ensure a fair comparison across offline baselines, we established the following constraints. Firstly, the batch size was set to 256 for all algorithms except for Decision Transformer. Secondly, the learning rate was set to 0.0003 for BC, IQL, ATQ, AWAC, EDAC, TD3, and SAC-N. The Decision Transformer [4] was excluded from these constraints due to its fundamentally different model architecture.

C.2 Implementation Details for ATQ

Source code. The source code will be made public upon acceptance. In the ATQ-based SAC method, we modified the policy gradient computation in the SAC as described in Algorithm 1. The backbone structure of SAC remains untouched in ATQ implementation. The update of critic networks follows the standard SAC procedure. It is worth noticing that we use different SAC backbones for online and offline RL. Standard SAC [14] backbone is used for online environments and SAC-N [1] backbone is used for offline datasets.

For the implementation of the ATQ method, the generation of QMC samples is by the Sobol sequence engine. After the generation of QMC samples, the process of approximating the policy gradient (8) is computed parallelly for all QMC samples.

Hyperparameters. The tables below shows the hyperparameters used in our experiments. The M indicates the logarithm of the maximum sample number for hardware limitations and cost tolerance. Table 2 shows the choices of hyperparameter β in Equation 11 for online environments. Table 3 shows the choices of hyperparameter β in Equation 11 for offline environments.

Table 2: Task-Specific Hyperparameters for Online Environments

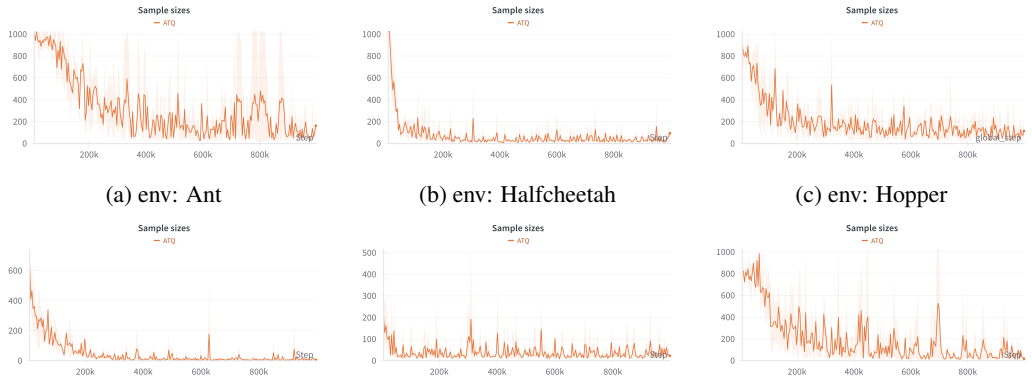
Task Name	β	**M**
Hopper	0.0003333	10
HalfCheetah	0.00015	10
Ant	0.0002	10
Walker2d	0.0003	10
HumanoidStandup	0.00001	10
Humanoid	0.001	10

Table 3: Task-Specific Hyperparameters for Offline Environments

Task Name	β	**M**
HalfCheetah-Medium	0.0000667	8
HalfCheetah-Medium-Replay	0.0000667	8
Hopper-Medium-Replay	0.0003	8
Walker-Medium-Replay	0.0003	8

C.3 Additional experiment results

We report some additional results to help illustrate the behavior of the ATQ-based SAC method. Figure 6 shows the sample sizes used in each iteration during training for online experiments. This



(a) env: Ant (b) env: Halfcheetah (c) env: Hopper

(d) env: Humanoid (e) env: Humanoidstandup (f) env: Walker

Figure 6: Sample sizes used per iteration for online different environments.

figure is directly related to Figure 2. From this figure, we can see the dynamic adjustment of sample sizes of the ATQ method.

D Limitations and future work

The limits of the current work can be summarized as follows. Firstly, additional experiments on a broader set of environments are encouraged to validate the generalizability of the proposed method. Secondly, the adaptive function used in our approach could be further refined to enhance its performance and applicability across different scenarios.