On Using Hamiltonian Monte Carlo Sampling for RL

Udari Madhushani, Biswadip Dey, Naomi Ehrich Leonard and Amit Chakraborty

***Abstract*— Q-Learning and other value function based reinforcement learning (RL) algorithms learn optimal policies from datasets of actions, rewards, and state transitions. However, generating independent and identically distributed (IID) data samples poses a significant challenge when the state transition dynamics are stochastic and high-dimensional; this is due to intractability of the associated normalizing integral. We address this challenge with Hamiltonian Monte Carlo (HMC) sampling since it offers a computationally tractable way to generate data for training RL algorithms in stochastic and high-dimensional contexts. We introduce *Hamiltonian Q-Learning* and use it to demonstrate, theoretically and empirically, that Q values can be learned from a dataset generated by HMC samples of actions, rewards, and state transitions. Hamiltonian Q-Learning also exploits underlying low-rank structure of the Q function using a matrix completion algorithm for reconstructing the Q function from Q value updates over a much smaller subset of state-action pairs. Thus, by providing an efficient way to apply Q-Learning in stochastic, high-dimensional settings, the proposed approach broadens the scope of RL algorithms for real-world applications.**

I. Introduction

In recent years, reinforcement learning (RL) has shown remarkable success with sequential decision-making tasks wherein an agent observes the current state of the environment, chooses an action, and receives a reward, before the environment transitions to a new state [1], [2]. RL has been applied to a variety of problems, such as control [3], robotics [4], resource allocation [5], and chemical process optimization [6]. However, existing model-free RL approaches typically perform well only when the environment has been explored long enough, and the algorithm has used a large enough number of samples [7], [8]. This is inherently challenging for high-dimensional stochastic problems.

Q-Learning is a model-free RL approach where an agent chooses its actions based on a policy defined by the state-action value function called the Q function [9], [10]. The performance of Q-Learning algorithms in stochastic settings depends strongly on the ability to access data samples that can provide accurate estimates of the expected Q values. As Q-Learning algorithms compute the expected Q values by calculating the sample mean of Q values over a set of independent and identically distributed (IID) samples, they assume access to a simulator that can generate IID samples according to the state transition probability. When the state transition probability distribution is high-dimensional, generating IID samples poses significant challenges: the lack of closed-form solutions and insufficiency of deterministic approximations of the normalizing integral. Because these prevent the use of existing RL methods, we were motivated to ask the question: *How can we develop value function based RL methods when generating IID samples is impractical?*

Crucial to developing such methods is identifying means to draw samples from an unnormalized distribution. Importance sampling offer techniques to draw samples from a distribution without computing the corresponding normalizing integral. Hamiltonian Monte Carlo (HMC) sampling is one such method, and thus allows samples to be generated from the unnormalized state transition distribution [11]. Our question is then: *How can we combine HMC sampling with Q-Learning to learn optimal policies for high-dimensional problems?*

In this paper, we introduce *Hamiltonian Q-Learning* to address the question. We show that Hamiltonian Q-Learning can infer optimal policies even when it calculates the expected Q values using HMC samples instead of IID samples. However, while HMC sampling overcomes the challenges associated with drawing IID samples in high dimensions, a large number of samples is still needed to learn the Q function because high-dimensional spaces often lead to a large number of state-action pairs, and thus high computational costs and high sample complexity. We address these issues by leveraging matrix completion techniques. It has been observed that formulating planning and control tasks as Q-Learning problems in a variety of contexts leads to low-rank structures in the associated Q matrix [12]–[14]. Since these systems naturally consist of a large number of states, exploiting the low-rank structure in the Q matrix in an informed way can reduce computational complexity. Hamiltonian Q-Learning uses matrix completion to reconstruct the Q matrix from a small subset of expected Q values making it data-efficient.

In related work, [12], [13] consider a model-free RL approach that exploits structures of the state-action value function. The work by [12] decomposes the Q matrix into a low-rank and sparse matrix model and uses matrix completion methods [15]–[17] to improve data-efficiency. A more recent work [13] has shown that incorporating low rank matrix completion methods to recover the Q matrix from a small subset of Q values can improve learning of optimal policies. The paper [14] extends this work by proposing a novel matrix estimation method and providing theoretical guarantees for the convergence to an ϵ-optimal Q function. Entropy regularization techniques penalize excessive randomness in the conditional distribution of actions for a given state and

This work was done while first author was interning at Siemens. Research also supported in part by ONR grant N00014-18-1- 2873.

U. Madhushani and N.E. Leonard are with Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA. {udarim, naomi}@princeton.edu

B. Dey and A. Chakraborty are with Siemens Corporation, 755 College Road East, Princeton, NJ 08536, USA. {biswadip.dey, amit.chakraborty}@siemens.com

provide an alternative means to implicitly exploit the low-dimensional structure of the value function [18]–[20].

The main contributions of this work are threefold. *First*, we introduce a modified Q-learning framework, called *Hamiltonian Q-learning*, which uses HMC sampling to provide a data-efficient approach for using Q-learning in real-world problems with high-dimensional state space and stochastic state transition. Integration of this sampling approach with matrix-completion enables us to update Q values for only a small subset of state-action pairs and reconstruct the complete Q matrix. *Second*, we provide theoretical guarantees that the error between the optimal Q function and the Q function computed by updating Q values using HMC sampling can be made arbitrarily small. This result holds even when only a small fraction of the Q values are updated using HMC samples and the rest are estimated using matrix completion. We also provide theoretical guarantee that the sampling complexity of our algorithm matches the mini-max sampling complexity proposed by [21]. *Third*, we apply Hamiltonian Q-learning to a high-dimensional problem (stabilizing a double pendulum on a cart) as well as to control tasks used as benchmarks in the machine learning literature (inverted pendulum, double integrator, cartpole, and acrobot). Our results show that the proposed approach becomes more effective with increase in state space dimension.

The rest of the paper is organized as follows. In Section II we provide a brief background and propose the Hamiltonian Q-Learning algorithm. We provide theoretical results and experimental results in Section III and IV, respectively. We provide a discussion and concluding remarks in Section V.

II. Hamiltonian Q-Learning

In this section we derive Hamiltonian Q-Learning, which we also present in pseudocode and denote as Algorithm 1.

Learning an optimal Q^* function through value iteration requires updating Q values of state-action pairs using a sum of the reward and a discounted expectation of Q values associated with next states. Although the state dynamics are assumed noisy, in this paper we assume that the reward r at time step t is a deterministic function of state-action pairs (s_t, a_t) at t. Our results can be extended to stochastic rewards by replacing the reward with its expectation.

The update Q^{t+1} of the Q value at time step $t+1$ is

$$Q^{t+1}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}\left(\max_a Q^t(s_{t+1}, a)\right), \quad (1)$$

where $\mathbb{E}$ denotes the expectation over the discrete probability measure $\mathbb{P}$ and $\lambda > 0$ is the discount factor. When the state space is high-dimensional and has large number of states in each dimension, we encounter two key challenges to learning the Q function: (*i*) difficulty in estimating the expectation in (1) due to the impracticality of generating IID samples and the high computational cost of exhaustive sampling; and (*ii*) a sample complexity that increases quadratically with the number of states and linearly with the number of actions.

A. HMC sampling for learning Q function

A number of importance-sampling methods [11], [22] have been developed for estimating the expectation of a function by drawing samples from the region with the dominant contribution to the expectation. HMC is one such importance-sampling method [11] that draws samples from the typical set, i.e., the region that maximizes probability mass, which provides the dominated contribution to the expectation. Since the decay in Q function is significantly smaller compared to the typical exponential or power law decay in transition probability functions, HMC provides a better approximation than $\mathbb{E}$ for the expectation of the Q value of the next states [13], [14]. Let $\mathcal{H}_t$ denote the set of HMC samples drawn at time step t. Then, in place of (1), we update the Q values as

$$Q^{t+1}(s_t, a_t) = r(s_t, a_t) + \frac{\gamma}{|\mathcal{H}_t|} \sum_{s \in \mathcal{H}_t} \max_a Q^t(s, a). \quad (2)$$

HMC for a smooth truncated target distribution. Recall that a region of states is a subset of a Euclidean space given as $[d_1^-, d_1^+] \times \ldots \times [d_{\mathcal{D}_s}^-, d_{\mathcal{D}_s}^+] \subset \mathbb{R}^{\mathcal{D}_s}$, where $\mathcal{D}_s$ is the dimension of the state space. Thus the main challenge to using HMC sampling is to define a smooth continuous target distribution $\mathcal{P}(s|s_t, a_t)$ on $\mathbb{R}^{\mathcal{D}_s}$ with a sharp decay at the boundary of the region of states [23], [24]. Here, we generate the target distribution by first defining the transition probability kernel from the conditional probability distribution defined on $\mathbb{R}^{\mathcal{D}_s}$ and then taking its product with a smooth cut-off function.

We first consider a probability distribution $\mathcal{P}(\cdot|s_t, a_t) : \mathbb{R}^{\mathcal{D}_s} \to \mathbb{R}$ such that the following holds:

$$\mathbb{P}(s|s_t, a_t) \propto \int_{s-\varepsilon}^{s+\varepsilon} \mathcal{P}(s|s_t, a_t) ds \quad (3)$$

for some arbitrarily small $\varepsilon > 0$. Then the target distribution can be defined as

$$\mathcal{P}(s|s_t, a_t) = \mathcal{P}(s|s_t, a_t) \times \prod_{i=1}^{\mathcal{D}_s} \left[\frac{1}{1 + \exp(-\kappa(d_i^+ - s^i))} \cdot \frac{1}{1 + \exp(-\kappa(s^i - d_i^-))} \right]. \quad (4)$$

Note that there exists a large $\kappa > 0$ such that if $s \in [d_1^-, d_1^+] \times \ldots \times [d_{\mathcal{D}_s}^-, d_{\mathcal{D}_s}^+]$ then $\mathcal{P}(s|s_t, a_t) \propto \mathbb{P}(s|s_t, a_t)$ and $\mathcal{P}(s|s_t, a_t) \approx 0$ otherwise. Let $\mu(s_t, a_t), \Sigma(s_t, a_t)$ be the mean and covariance of the transition probability kernel. Here we consider transition probability kernels of the form

$$\mathbb{P}(s|s_t, a_t) \propto \exp\left(-\frac{1}{2}(s - \mu(s_t, a_t))^T \Sigma^{-1}(s_t, a_t)(s - \mu(s_t, a_t)) \right). \quad (5)$$

Then from (3) the corresponding mapping can be given as a multivariate Gaussian $\mathcal{P}(s|s_t, a_t) = \mathcal{N}(\mu(s_t, a_t), \Sigma(s_t, a_t))$. From (4) it follows that the target distribution is

$$\mathcal{P}(s|s_t, a_t) = \mathcal{N}(\mu(s_t, a_t), \Sigma(s_t, a_t)) \prod_{i=1}^{\mathcal{D}_s} \frac{1}{1 + \exp(-\kappa(d_i^+ - s^i))} \frac{1}{1 + \exp(-\kappa(s^i - d_i^-))}. \quad (6)$$

Choice of potential energy and kinetic energy.

For brevity of notation we drop the explicit dependence of $\mathcal{P}(\cdot)$ on (s_t, a_t) and denote the target distribution as $\mathcal{P}(s)$ defined over the Euclidean space $\mathbb{R}^{\mathcal{D}_s}$. We choose the potential energy as $U(s) = -\log(\mathcal{P}(s))$ and kinetic energy $K(v, s) = -\log \mathcal{P}(v|s) = \frac{1}{2} v^T M^{-1} v$ where M^{-1} is the covariance of the target distribution and v is the momentum variable. For additional details on HMC sampling we refer readers to [11].

B. Q-Learning with HMC and matrix completion

In this work we consider problems with a high-dimensional state space and large number of distinct states along individual dimensions. Although these problems admit a large Q matrix, we can exploit low rank structure of the Q matrix to further improve the sample efficiency.

At each time step t we randomly sample a subset Ω_t of state-action pairs (each state-action pair is sampled independently with some probability p) and update the Q function for state-action pairs in Ω_t. Let $\widehat{Q}^{t+1}$ be the updated Q matrix at time t. Then from (2) we have

$$\widehat{Q}^{t+1}(s_t, a_t) = r(s_t, a_t) + \frac{\gamma}{|\mathcal{H}_t|} \sum_{s \in \mathcal{H}_t} \max_a Q^t(s, a), \quad (7)$$

for any $(s_t, a_t) \in \Omega_t$. We recover the complete matrix Q^{t+1} by using matrix completion as follows:

$$\begin{aligned} Q^{t+1} = \underset{\widetilde{Q}^{t+1} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}}{\arg\min} \quad & \|\widetilde{Q}^{t+1}\|_* \\ \text{subject to } & \jmath_{\Omega_t}\left(\widetilde{Q}^{t+1}\right) = \jmath_{\Omega_t}\left(\widehat{Q}^{t+1}\right) \end{aligned} \quad (8)$$

where $|\mathcal{S}| \times |\mathcal{A}|$ is total number of state-action pairs and $\jmath_{\Omega_t}$ is the observation operator, i.e. $\jmath_{\Omega}(x) = x$ if $x \in \Omega$ and zero otherwise. Similar to the approach used in [13], we approximate the rank of the Q matrix as the minimum number of singular values needed to capture 99% of its nuclear norm.

Algorithm 1 Hamiltonian Q-Learning

Inputs: Discount factor γ; Range of state space; Time horizon T;
Initialization: Randomly initialize Q^0
for $t = 1$ **to** T **do**
Step 1: Randomly sample a subset of state-action pairs Ω_t
Step 2: **HMC sampling phase** - Sample a set of next states $\mathcal{H}_t$ according to the target distribution defined in (4)
Step 3: **Update phase** - For all $(s_t, a_t) \in \Omega_t$
$\widehat{Q}^{t+1}(s_t, a_t) = r(s_t, a_t) + \frac{\gamma}{|\mathcal{H}_t|} \sum_{s \in \mathcal{H}_t} \max_a Q^t(s, a)$
Step 4: **Matrix Completion phase**
$Q^{t+1} = \underset{\widetilde{Q}^{t+1} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}}{\arg\min} \|\widetilde{Q}^{t+1}\|_*$
subject to $\jmath_{\Omega_t}\left(\widetilde{Q}^{t+1}\right) = \jmath_{\Omega_t}\left(\widehat{Q}^{t+1}\right)$
end for

III. Convergence, Boundedness and Sampling Complexity

In this section we provide theoretical results on HMC Q-Learning. Because of space constraints we provide sketches of proofs and refer to [25] for details of proofs.

First, we introduce the following *regularity assumptions*:

(**A1**) The state space $\mathcal{S} \subseteq \mathbb{R}^{\mathcal{D}_s}$ and the action space $\mathcal{A} \subseteq \mathbb{R}^{\mathcal{D}_a}$ are compact subsets.

(**A2**) The reward function is bounded, i.e., $r(s, a) \in [R_{\min}, R_{\max}]$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$.

(**A3**) The optimal value function Q^* is C-Lipschitz, i.e.

$$\left|Q^*(s, a) - Q^*(s', a')\right| \leq C\Big(||s - s'||_F + ||a - a'||_F\Big)$$

where $||\cdot||_F$ is the Frobenius norm (which is the same as the Euclidean norm for vectors).

We provide theoretical guarantees that Hamiltonian Q-Learning converges to an ϵ-optimal Q function with $\widetilde{O}\left(\frac{1}{\epsilon^{\mathcal{D}_s + \mathcal{D}_a + 2}}\right)$ number of samples. This matches the minimax lower bound $\Omega\left(\frac{1}{\epsilon^{\mathcal{D}_s + \mathcal{D}_a + 2}}\right)$ proposed in [21]. First we define a family of ϵ-optimal Q functions as follows.

Definition 1 (**ϵ-optimal Q functions**). *Let Q^* be the unique fixed point of the Bellman optimality equation: $(\mathcal{T}Q)(s', a') = \sum_{s \in \mathcal{S}} \mathbb{P}(s|s', a') \left(r(s', a') + \gamma \max_a Q(s, a)\right)$, $\forall (s', a') \in \mathcal{S} \times \mathcal{A}$ where $\mathcal{T}$ denotes the Bellman operator. Then, under update rule (1), the Q function almost surely converges to the optimal Q^*. We define ϵ-optimal Q functions as the family of functions $\mathbf{Q}_\epsilon$ such that $\|Q' - Q^*\|_\infty \leq \epsilon$ whenever $Q' \in \mathbf{Q}_\epsilon$.*

As $\|Q' - Q^*\|_\infty = \max_{(s,a) \in \mathcal{S} \times \mathcal{A}} \|Q'(s, a) - Q^*(s, a)\|$, any ϵ-optimal Q function is element wise ϵ-optimal. Our next result shows that under the HMC sampling rule given in Step 3 of Hamiltonian Q-Learning (Algorithm 1), the Q function converges to the family of ϵ-optimal Q functions.

Theorem 1 (**Convergence of Q function under HMC Sampling**). *Let $\mathcal{T}$ be an optimality operator under HMC sampling given as $(\mathcal{T}Q)(s', a') = r(s', a') + \frac{\gamma}{|\mathcal{H}|} \sum_{s \in \mathcal{H}} \max_a Q(s, a)$, $\forall (s', a') \in \mathcal{S} \times \mathcal{A}$, where $\mathcal{H}$ is a subset of next states sampled using HMC from the target distribution given in (4). Then, under update rule (2) and for any given $\epsilon \geq 0$, there exists $n_{\mathcal{H}}, t' > 0$ such that $\|Q^t - Q^*\|_\infty \leq \epsilon$, $\forall t \geq t'$.*

Proof. (*sketch*) We follow a similar approach to the proof of Q-function convergence under exhaustive sampling, with a key modification that accounts for the error incurred by HMC sampling. We note that the Q-function error under HMC sampling can be upper bounded by the summation of (*i*) the Q-function error under exhaustive sampling and (*ii*) the error between the empirical average under HMC sampling and the expectation under exhaustive sampling. Thus, when the Q-function is Lipschitz, from the central limit theorem for HMC sampling we can upper bound the cumulative error induced by the second term using a constant. □

The next theorem shows that the Q matrix estimated with a suitable matrix completion technique lies in the ϵ-neighborhood of the corresponding Q function obtained with exhaustive sampling.

Theorem 2 (Bounded Error under HMC with Matrix Completion). *Let* $Q_{\mathcal{E}}^{t+1}(s_t, a_t) = r(s_t, a_t) + \gamma \sum_{s \in \mathcal{S}} \mathbb{P}(s|s_t, a_t) \max_a Q_{\mathcal{E}}^t(s, a)$, $\forall (s_t, a_t) \in \mathcal{S} \times \mathcal{A}$ *be the update rule under exhaustive sampling, and* Q^t *be the* Q *function updated according to Hamiltonian* Q*-Learning (7)-(8). Then, for any given* $\tilde{\epsilon} \geq 0$*, there exists* $n_{\mathcal{H}} = \min_\tau |\mathcal{H}_\tau|, t' > 0$*, such that* $\|Q^t - Q_{\mathcal{E}}^t\|_\infty \leq \tilde{\epsilon}$, $\forall t \geq t'$.

Proof. (*sketch*) Due to boundedness under matrix completion, the error between Q functions updated according to Hamiltonian Q-Learning and exhaustive sampling can be upper bounded using summation of (*i*) error between updated $\widehat{Q}^t$ and optimal function Q^* and (*ii*) error between updated function $Q_{\mathcal{E}}^t$ under exhaustive sampling and optimal function Q^*. The proof follows from upper bounding the first term using matrix completion boundedness results and the second term using Theorem 1. □

Finally we provide guarantees on the sampling complexity of Hamiltonian Q-Learning (Algorithm 1).

Theorem 3. (Sampling complexity of Hamiltonian Q-Learning) *Let* $\mathcal{D}_s$, $\mathcal{D}_a$ *be the dimension of state space and action space, respectively. Consider the Hamiltonian* Q*-Learning algorithm presented in Algorithm 1. Then, under a suitable matrix completion method, the* Q *function converges to the family of* ϵ*-optimal* Q *functions with* $\widetilde{O}\left(\epsilon^{-(\mathcal{D}_s + \mathcal{D}_a + 2)}\right)$ *number of samples.*

Proof. (*sketch*) Let T_ϵ be the time step such that the learned Q function under Hamiltonian Q-Learning is ϵ-optimal. Then, the number of samples required by Hamiltonian Q-Learning to learn an ϵ-optimal Q function is $\sum_{t=1}^{T_\epsilon} |\Omega_t||\mathcal{H}_t|$. We first prove results on the sample size $|\Omega_t|$ required to bound the error incurred due to matrix completion. Then we prove results on the sample size $|\Omega_t|$ required to bound the error incurred by approximating the expectation of the next state using HMC samples. The final result follows from combining these results with the convergence and boundedness results obtained in Theorems 1 and 2. □

IV. Numerical Experiments

We illustrate convergence and sample efficiency of Hamiltonian Q-Learning using a high-dimensional control system and four benchmark control tasks. Because Lipschitz convergence in Frobenius norm of the Q function implies convergence in the infinity norm, we use the Frobenius norm of the difference between the learned Q function and optimal Q^* to illustrate that Hamiltonian Q-Learning converges to an ϵ-optimal Q function. See [25] for dynamic equations for all systems.

A. Empirical Evaluation for a High-Dimensional System

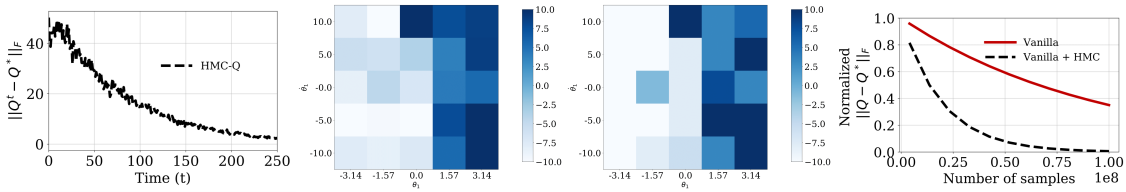
Experimental setup for a double pendulum on a cart: Let $x, \dot{x}$ denote the position and velocity of the cart and $\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2$ denote the joint angles and angular velocities of the poles. We define the 6-dimensional state of the cart-pole system as: $s = (x, \dot{x}, \theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2)$ where $x \in [-2.4, 2.4]$, $\dot{x} \in [-3.5, 3.5]$, and $\theta_i \in [-\pi, \pi]$, $\dot{\theta}_i \in [-3.0, 3.0]$ for $i = 1, 2$. Also, we define the range of the scalar action as $a \in [-10, 10]$. Each state space dimension is discretized into 5 distinct values and the action space into 10 distinct values. This leads to a Q matrix of size 15625×10. We consider that the probabilistic state transition is governed by (5) with a Σ which ensures that the range of the state space along direction i is approximately equal to $6\sqrt{\Sigma_i}$. To stabilize the pendulum to an upright position, we define the reward function as $r(s, a) = \cos^4(15\theta_1) + \cos^4(15\theta_2)$. After initializing the Q matrix using randomly chosen values from $[0, 2]$, we sample state-action pairs with probability $p = 0.2$ at each iteration.

Results: Figure 1(a) shows the change in the Frobenius norm of the difference between the learned Q function and optimal Q^*, thereby illustrating that Hamiltonian Q-Learning converges to an ϵ optimal Q function. Note that under exhaustive sampling we use 15625 samples for each update. However, Hamiltonian Q-Learning uses only 200 samples for each update. As it is difficult to visualize policy heat maps for a 6-dimensional state space, we show results for the first two dimensions (i.e., θ_1 and $\dot{\theta}_1$) while keeping the rest fixed (i.e., $\theta_2 = 0$, $\dot{\theta}_2 = 0$, $x = -1.2$, and $\dot{x} = 3.5$). The heat maps shown in Figures 1(b) and 1(c) illustrate that the policy heat map for Hamiltonian Q-Learning is close to the one from Q-Learning with exhaustive sampling. We also show that the sample efficiency of Q-Learning can be significantly improved by incorporating Hamiltonian Q-Learning. Figure 1(d) shows how the Frobenius norm of the difference between the learned Q function and the optimal Q^*, normalized by its maximum value, varies with increase in the number of samples. The solid red line shows the accuracy for Q-Learning with exhaustive sampling and the dashed black line shows the same for Hamiltonian Q-Learning. These results show that Hamiltonian Q-Learning converges to an ϵ-optimal Q function with significantly fewer samples than Q-Learning with exhaustive sampling.

B. Empirical Evaluation for Low-Dimensional Systems

Experimental setup: Here we investigate the applicability of Hamiltonian Q-Learning in low-dimensional spaces where IID samples are available, and compare its performance against state-of-the-art algorithms on four benchmark control tasks (inverted pendulum, double integrator, cartpole, and acrobot). The dynamics of the inverted pendulum and double integrator evolve on a 2-dimensional state space, and the cartpole and acrobot evolve on a 4-dimensional state space. We discretize each state space dimension of the inverted pendulum and double integrator into 25 distinct values, and each state space dimension of the cartpole and acrobot into 5 distinct values. The action variable associated with all four control tasks is scalar, and we discretize each action space into 10 distinct values. The size of the Q matrix is 625×10.

Results: Figure 2 shows that the Frobenius norm of the difference between the learned Q function and optimal Q^* can achieve a much lower value when HMC samples are used instead of IID samples. This illustrates that Hamiltonian Q-Learning achieves better convergence than Q-Learning with IID sampling. Note that, under exhaustive sampling



(a) Q Function Convergence (b) Exhaustive Sampling (c) HMC Sampling (d) Sample efficiency

Fig. 1: Figure 1(a) illustrates convergence of the Q function learned with Hamiltonian Q-Learning to an ϵ-optimal Q function. Figure 1(b) and 1(c) show policy heat maps for Q-Learning with exhaustive sampling and Hamiltonian Q-Learning, respectively $(x = -1.2, \dot{x} = 1.75, \theta_2 = \pi/4, \dot{\theta}_2 = 1.5)$. Figure 1(d) shows the change in the normalized value of the Frobenius norm with the number of samples, for both Q-Learning with exhaustive sampling (Vanilla) and Hamiltonian Q-Learning (Vanilla + HMC).

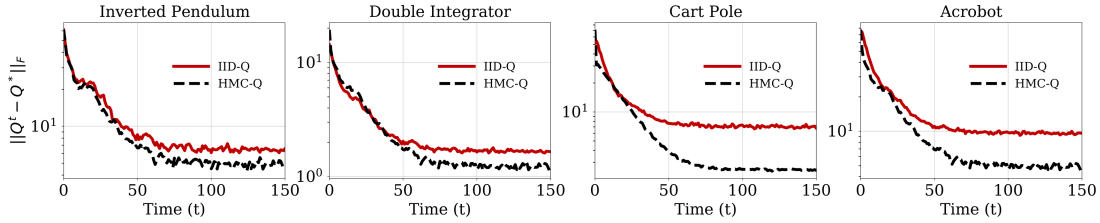


Fig. 2: A comparison of convergence of Q function with Hamiltonian Q-Learning and Q-Learning with IID sampling.

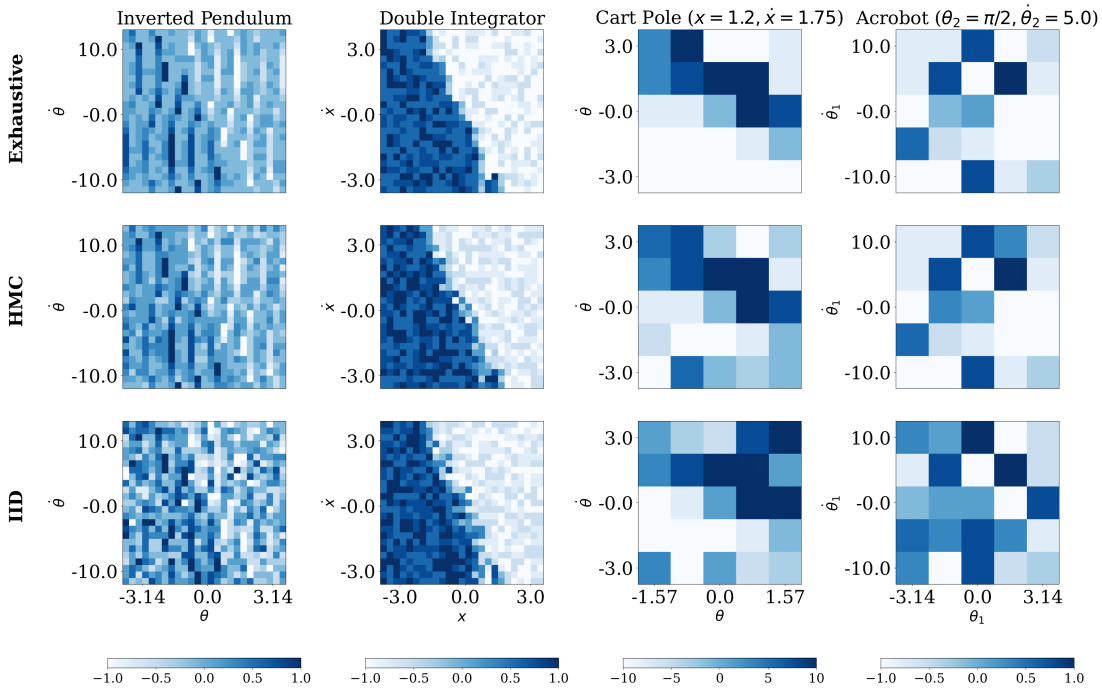


Fig. 3: Policy heatmaps for Q-Learning with exhaustive sampling, Hamiltonian Q-Leaning and Q-Learning with IID sampling. The color in each cell corresponds to the value of optimal action at the corresponding state.

we use 625 samples for each update, whereas learning with IID sampling and Hamiltonian Q-Learning require only 100 samples for each update. Figure 3 shows policy heatmaps for Q-Learning with exhaustive sampling, Hamiltonian Q-Learning and Q-Learning with IID sampling. Our results show that the policy heatmaps associated from Hamitonian Q-Learning are closer to policy heatmaps obtained from Q-Learning with exhaustive sampling. Figure 4 illustrates how the normalized Frobenius norm of the difference between the learned Q function and the optimal Q^* varies with increase in the number of samples. The solid red lines correspond to Q-Learning with exhaustive sampling and the dashed black lines correspond to Hamiltonian Q-Learning. These results show that Hamiltonian Q-Learning can achieve the same level of accuracy with significantly fewer samples.

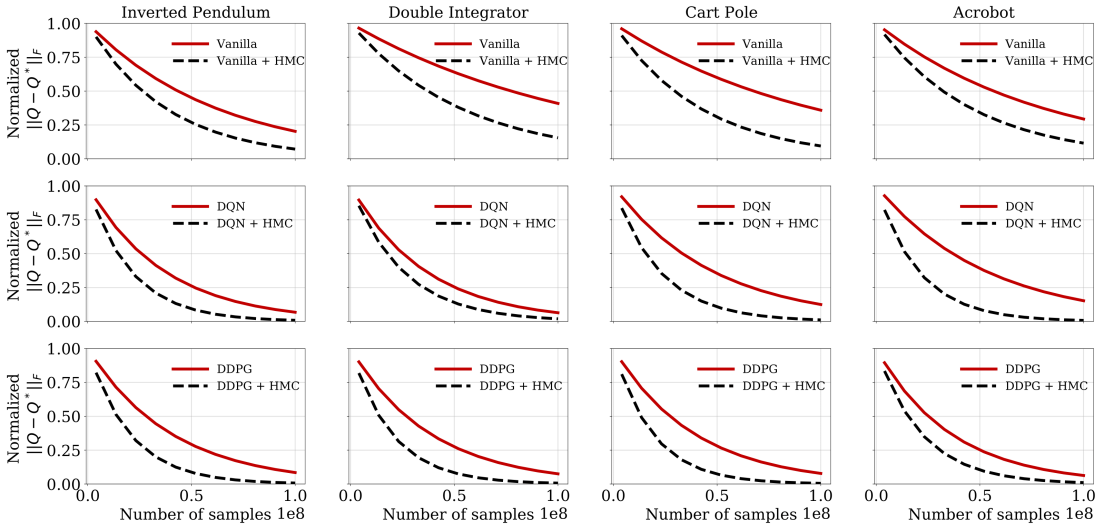


Fig. 4: Frobenius norm vs number of samples of Q function for Q-Learning with exhaustive sampling (Vanilla), Q-Learning with HMC sampling (Vanilla + HMC), Deep Q-Network (DQN) with exhaustive sampling, DQN with HMC sampling, Deep Deterministic Policy Gradient (DDPG) with exhaustive sampling and DDPG with HMC sampling. Red solid curve corresponds to exhaustive sampling and back dotted curve corresponds to HMC sampling.

V. Discussion and Conclusion

In this paper we have introduced *Hamiltonian Q-Learning*, a new model-free RL framework that can be utilized to obtain optimal policies in high-dimensional spaces, where generating IID samples is impractical. We showed, both theoretically and empirically, that the proposed approach can learn accurate estimates of the optimal Q function with many fewer samples as compared to exhaustive sampling. Further, we illustrated that Hamiltonian Q-Learning can be used to improve sample efficiency of state-of-the-art algorithms in low dimensional spaces as well. Leveraging these results, future works will investigate how HMC sampling based methods can improve sample efficiency in multi-agent Q-learning, a system naturally very high-dimensional, with agents coupled through both action and reward.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

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

[3] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016, pp. 1329–1338.

[4] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[5] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, 2016, pp. 50–56.

[6] Z. Zhou, X. Li, and R. N. Zare, "Optimizing chemical reactions with deep reinforcement learning," *ACS Central Science*, vol. 3, no. 12, pp. 1337–1344, 2017.

[7] S. Kamthe and M. Deisenroth, "Data-efficient reinforcement learning with probabilistic model predictive control," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 1701–1710.

[8] Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, and V. Sindhwani, "Data efficient reinforcement learning for legged robots," in *Conference on Robot Learning*. PMLR, 2020, pp. 1–10.

[9] C. J. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, Cambridge, UK, 1989.

[10] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[11] M. Betancourt, "A conceptual introduction to hamiltonian monte carlo," *arXiv preprint arXiv:1701.02434*, 2017.

[12] H. Y. Ong, "Value function approximation via low-rank models," *arXiv:1509.00061*, 2015.

[13] Y. Yang, G. Zhang, Z. Xu, and D. Katabi, "Harnessing structures for value-based planning and reinforcement learning," in *International Conference on Learning Representations*, 2020.

[14] D. Shah, D. Song, Z. Xu, and Y. Yang, "Sample efficient reinforcement learning via low-rank matrix estimation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 092–12 103, 2020.

[15] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[16] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, 2012.

[17] Y. Chen and Y. Chi, "Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization," *IEEE Signal Processing Magazine*, vol. 35, no. 4, pp. 14–31, 2018.

[18] Z. Ahmed, N. Le Roux, M. Norouzi, and D. Schuurmans, "Understanding the impact of entropy on policy optimization," in *International Conference on Machine Learning*. PMLR, 2019, pp. 151–160.

[19] W. Yang, X. Li, and Z. Zhang, "A regularized approach to sparse optimal policy in reinforcement learning," in *Advances in Neural Information Processing Systems*, 2019, pp. 5940–5950.

[20] E. Smirnova and E. Dohmatob, "On the convergence of smooth regularized approximate value iteration schemes," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[21] A. B. Tsybakov, *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[22] J. S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statistics and Computing*, vol. 6, no. 2, pp. 113–119, 1996.

[23] K. Yi and F. Doshi-Velez, "Roll-back hamiltonian monte carlo," *arXiv preprint arXiv:1709.02855*, 2017.

[24] A. Chevallier, S. Pion, and F. Cazals, "Hamiltonian Monte Carlo with boundary reflections, and application to polytope volume calculations," 2018.

[25] U. Madhushani, B. Dey, N. E. Leonard, and A. Chakraborty, "On using hamiltonian monte carlo sampling for reinforcement learning," *arXiv preprint arXiv:2011.05927*, 2020.