On Using Hamiltonian Monte Carlo Sampling for Reinforcement Learning Problems in High-dimension

Anonymous Author(s)
Affiliation
Address
email

Abstract

Value function based reinforcement learning (RL) algorithms, for example, Q-learning, learn optimal policies from datasets of actions, rewards, and state transitions. However, when the underlying state transition dynamics are stochastic and evolve on a high-dimensional space, generating independent and identically distributed (IID) data samples for creating these datasets poses a significant challenge due to the intractability of the associated normalizing integral. In these scenarios, Hamiltonian Monte Carlo (HMC) sampling offers a computationally tractable way to generate data for training RL algorithms. In this paper, we introduce a framework, called *Hamiltonian Q-Learning*, that demonstrates, both theoretically and empirically, that Q values can be learned from a dataset generated by HMC samples of actions, rewards, and state transitions. Furthermore, to exploit the underlying low-rank structure of the Q function, Hamiltonian Q-Learning uses a matrix completion algorithm for reconstructing the updated 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.

1 Introduction

In recent years, reinforcement learning has shown remarkable success with sequential decision-making tasks wherein an agent, after observing the current state of the environment, chooses an action to receive a reward, and subsequently, the environment transitions to a new state [1, 2]. RL has been applied to a variety of problems, such as automatic 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 number of samples in the process [7, 8]. 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, i.e., the Q function [9, 10]. The performance of Q-learning algorithms depends strongly on the ability to access data samples, which can provide accurate estimates of the expected Q values.

As these algorithms compute the expected Q values by calculating the sample mean of Q values over a set of IID samples, they assume access to a simulator that can generate IID samples according to the state transition probability. However, when the state transition probability distribution is high-dimensional, generating IID samples poses a significant challenge due to - (i) lack of closed-form solutions, and (ii) insufficiency of deterministic approximations, of the normalizing integral, preventing the utilization of existing RL methods. This motivated us to ask - *How can we develop value function based RL methods when generating IID samples is impractical?*

A crucial step in developing such methods is identifying means to draw samples from an unnormalized distribution. Importance sampling methods offer techniques to draw samples from a distribution

Submitted to 7th Deep Reinforcement Learning Workshop (@NeurIPS 2021), Sydney, Australia.

without computing the corresponding normalizing integral. Hamilton Monte Carlo (HMC) sampling is one such method; it allows one to generate samples from the unnormalized state transition distribution [11]. Equipped with HMC, we attempt to answer the following question: *How can we combine HMC sampling with Q-Learning to learn optimal policies for high-dimensional problems?*

In this work, we introduce *Hamiltonian Q-Learning* to answer this 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. Now, even though HMC samples overcome 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. We address this issue by leveraging matrix completion techniques. It has been observed that formulating planning and control tasks in a variety of problems, such as video games (e.g., Atari games) and classical control problems (e.g., simple pendulum, cart pole) as Q-Learning problems leads to low-rank structures in the Q matrix associated with the problem [12, 13, 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 enable further reduction in the 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.

The three main contributions of this work are threefold. *First*, we introduce a modified Q-learning framework, called *Hamiltonian Q-learning*, which uses HMC sampling for efficient computation of the Q values. This innovation, by proposing to sample Q values from the region with the dominant contribution to the expectation of discounted reward, provides a data-efficient approach for using Q-learning in real-world problems with high-dimensional state space and probabilistic 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 [15]. *Finally*, we apply Hamiltonian Q-learning to a high-dimensional problem (in particular, the problem of stabilizing a double pendulum on a cart) as well as to benchmark control tasks (inverted pendulum, double integrator, cartpole, and acrobot). Our results show that the proposed approach becomes more effective with increase in state space dimension.

Related Work: The last decade has witnessed a growing interest in improving sample efficiency in RL methods by exploiting emergent global structures from underlying system dynamics. [7, 16, 17, 18] have proposed model-based RL methods that improve sample efficiency by explicitly incorporating prior knowledge about state transition dynamics of the underlying system. [19, 20, 21] propose Baysean methods to approximate the Q function. [12, 13] consider a model-free RL approach that exploit structures of 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 [22, 23, 24] to improve sample efficiency. A more recent work [13] has shown that incorporating low rank matrix completion methods to recover Q matrix from a small subset of Q values can improve learning of optimal policies. At each time step the agent chooses a subset of state-action pairs and update the corresponding Q value using the Bellman optimally equation that considers a discounted average between reward and expectation of the Q values of next states. [14] extends this work by proposing a novel matrix estimation method and providing theoretical guarantees for the convergence to a ϵ-optimal Q function. On the other hand, entropy regularization techniques penalize excessive randomness in the conditional distribution of actions for a given state and provide an alternative means to implicitly exploit the underlying low-dimensional structure of the value function [25, 26, 27]. [28] has proposed an approach that samples a whole episode and then updates values in a recursive, backward manner.

2 Preliminary Concepts

In this section, we provide a brief background on Q-Learning, HMC sampling and matrix completion, as well as introduce the mathematical notations. In this paper, $|\mathbb{Z}|$ denotes the cardinality of a set $\mathbb{Z}$. Moreover, $\mathbb{R}$ represent the real line and A^T denotes the transpose of matrix A.

2.1 Q-Learning

Markov Decision Process (MDP) is a mathematical formulation that captures salient features of sequential decision making [29]. In particular, a *finite MDP* is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma)$, where $\mathcal{S}$ is the finite set of system states, $\mathcal{A}$ is the finite set of actions, $\mathbb{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a bounded reward function, and $\gamma \in [0, 1)$ is a discounting factor. Without loss of generality, states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$ can be assumed to be $\mathcal{D}_s$-dimensional and $\mathcal{D}_a$-dimensional real vectors, respectively. Moreover, by letting s^i denote the ith element of a state vector, we define the range of state space in terms of the following intervals $[d_i^-, d_i^+]$ such that $s^i \in [d_i^-, d_i^+]\ \forall i \in \{1, \ldots, \mathcal{D}_s\}$. At each time $t \in \{1, \ldots, T\}$ over the decision making horizon, an agent observes the state of the environment $s_t \in \mathcal{S}$ and takes an action a_t according to some policy π which maximizes the discounted cumulative reward. Once this action has been executed, the agent receives a reward $r(s_t, a_t)$ from the environment and the state of the environment changes to s_{t+1} according to the transition probability kernel $\mathbb{P}\left(\cdot|s_t, a_t\right)$. The Q function, which represents the expected discounted reward for taking a specific action at the current time and following the policy thereafter, is defined as a mapping from the space of state-action pairs to the real line, i.e. $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Then, by letting Q^t represent the Q matrix at time t, i.e. the tabulation of Q function over all possible state-action pairs associated with the finite MDP, we can express the Q value iteration over time steps as

$$Q^{t+1}(s_t, a_t) = \sum_{s \in \mathcal{S}} \mathbb{P}\left(s|s_t, a_t\right) \Big(r(s_t, a_t) + \gamma \max_a Q^t(s, a) \Big). \tag{1}$$

Under this update rule, the Q function converges to its optimal value Q^* [30]. To compute this sum (1) over possible next states, existing methods rely on either exhaustive sampling or a simulator generating IID samples. However they fail in high-dimensional spaces due to prohibitively high computational cost associated with calculating the normalizing integral of state transition distribution.

2.2 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo is an efficient sampling approach for drawing samples from probability distributions known up to a constant, i.e., unnormalized distributions. It offers faster convergence than Markov Chain Monte Carlo (MCMC) sampling [11, 31, 32, 33]. To draw samples from a smooth target distribution $\mathcal{P}(s)$, which is defined on the Euclidean space and assumed to be known up to a constant, HMC extends the target distribution to a joint distribution over the target variable s (viewed as position within the HMC context) and an auxiliary variable v (viewed as momentum within the HMC context). We define the Hamiltonian of the system as $H(s, v) = -\log \mathcal{P}(s, v) = -\log \mathcal{P}(s) - \log \mathcal{P}(v|s) = U(s) + K(v, s)$, where $U(s) \triangleq -\log \mathcal{P}(s)$ and $K(v, s) \triangleq -\log \mathcal{P}(v|s) = \frac{1}{2} v^T M^{-1} v$ represent the potential and kinetic energy, respectively, and M is a suitable choice of the mass matrix.

HMC sampling method consists of the following *three* steps $-$ (i) a new momentum variable v is drawn from a fixed probability distribution, typically a multivariate Gaussian; (ii) then a new proposal (s', v') is obtained by generating a trajectory that starts from (s, v) and obeys Hamiltonian dynamics, i.e. $\dot{s} = \frac{\partial H}{\partial v}, \dot{v} = -\frac{\partial H}{\partial s}$; and (iii) finally this new proposal is accepted with probability $\min \left\{1, \exp\left(H(s, v) - H(s', -v')\right)\right\}$ following the Metropolis–Hastings acceptance/rejection rule.

Thus HMC sampling offers a way to draw samples from unnormalized transition distributions often encountered in high-dimensional state spaces. However, since such problems often consist of a large number of state-action pairs, learning the Q function still requires a large number of samples. This leads to poor sample efficiency.

2.3 Low-rank Structure in Q-learning and Matrix Completion

When a matrix is low-rank or has a sparse structure, matrix completion methods can reconstruct it accurately from a small subset of entries. Prior work [34, 35, 12, 14] on value function approximation based approaches for RL has implicitly assumed that the state-action value functions are low-dimensional and used various basis functions to represent them, e.g. CMAC, radial basis function, etc. This can be attributed to the fact that the underlying state transition and reward function are often endowed with some structure. More recently, [13] provide empirical guarantees that the Q-matrices for benchmark Atari games and classical control tasks exhibit low-rank structure.

Therefore, using matrix completion techniques [36, 24] to recover $Q \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ from few observed Q values constitutes a viable approach towards improving sample efficiency. As low-rank matrix structures can be recovered by constraining the nuclear norm (i.e., the sum of its singular values), the Q matrix can be reconstructed from its observed values ($\hat{Q}$) by solving

$$\begin{aligned} Q = \underset{\widetilde{Q} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}}{\arg\min} \quad & \|\widetilde{Q}\|_* \\ \text{subject to} \quad & \mathcal{J}_\Omega(\widetilde{Q}) = \mathcal{J}_\Omega(\hat{Q}) \end{aligned} \tag{2}$$

where $\|\cdot\|_*$ denotes the nuclear norm, Ω is the observed set of elements, and $\mathcal{J}_\Omega$ is the observation operator, i.e. $\mathcal{J}_\Omega(x) = x$ if $x \in \Omega$ and zero otherwise.

3 Hamiltonian Q-Learning

A large class of real world sequential decision making problems - for example, board/video games, control of a robot's movement, and portfolio optimization - involves high-dimensional state spaces and often has large number of distinct states along each individual dimension. As using a Q-Learning based approach to train RL-agents for these problems typically requires tens to hundreds of millions of samples [1, 37], there is a strong need for sample efficient algorithms for Q-Learning. In addition, state transition in such systems is often probabilistic in nature; even when the underlying dynamics of the system is inherently deterministic; presence of external disturbances and parameter variations/uncertainties lead to probabilistic state transitions.

Learning an optimal Q^* function through value iteration methods 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. In this work, we assume the reward to be a deterministic function of state-action pairs. However, when the reward is stochastic, these results can be extended by replacing the reward with its expectation. Subsequently, we can express (1) as

$$Q^{t+1}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}\left(\max_a Q^t(s, a)\right), \tag{3}$$

where $\mathbb{E}$ denotes the expectation over the discrete probability measure $\mathbb{P}$. When the underlying state space is high-dimensional and has large number of states, we encounter two key challenges while attempting to learn the Q function: (*i*) difficulty in estimating the expectation in (3) due to high computational cost of exhaustive sampling and impracticality of generating IID samples; and (*ii*) a sample complexity that increases quadratically with the number of states and linearly with the number of actions.

To the best of our knowledge, *Hamiltonian Q-Learning* offers the first solution to this problem by combining *HMC sampling* and *matrix completion* that overcome the first and the second challenge, respectively.

3.1 HMC sampling for learning Q function

A number of importance-sampling methods [38, 31] 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 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 decays in transition probability function, HMC provides a better approximation for the expectation of the Q value of the next states [13, 14]. Then by letting $\mathcal{H}_t$ denote the set of HMC samples drawn at time step t, we update the Q values as:

$$Q^{t+1}(s_t, a_t) = r(s_t, a_t) + \frac{\gamma}{|\mathcal{H}_t|} \sum_{s \in \mathcal{H}_t} \max_a Q^t(s, a). \tag{4}$$

HMC for a smooth truncated target distribution: Recall that region of states is a subset of a Euclidean space given as $s \in [d_1^-, d_1^+] \times \ldots \times [d_{\mathcal{D}_s}^-, d_{\mathcal{D}_s}^+] \subset \mathbb{R}^{\mathcal{D}_s}$. Thus the main challenge to using HMC sampling is to define a smooth continuous target distribution $\mathcal{P}(s|s_t, a_t)$ which is defined on

$\mathbb{R}^{\mathcal{D}_s}$ with a sharp decay at the boundary of the region of states [39, 40]. In this work, 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 multiplying it 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

$$\mathbb{P}(s|s_t, a_t) \propto \int_{s-\varepsilon}^{s+\varepsilon} \mathcal{P}(s|s_t, a_t) ds \tag{5}$$

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

$$\mathscr{P}(s|s_t, a_t) = \mathcal{P}(s|s_t, a_t) \prod_{i=1}^{\mathcal{D}_s} \left[\frac{1}{1 + \exp(-\kappa(d_i^+ - s^i))} \cdot \frac{1}{1 + \exp(-\kappa(s^i - d_i^-))} \right]. \tag{6}$$

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 $\mathscr{P}(s|s_t, a_t) \propto \mathbb{P}(s|s_t, a_t)$ and $\mathscr{P}(s|s_t, a_t) \approx 0$ otherwise. Let $\mu(s_t, a_t), \Sigma(s_t, a_t)$ be the mean and covariance of the transition probability kernel. In this paper we consider transition probability kernels of the form

$$\mathbb{P}(s|s_t, a_t) \propto \exp\left(-\frac{1}{2}(s - \mu(s_t, a_t))^T \Sigma^{-1}(s_t, a_t)(s - \mu(s_t, a_t)) \right). \tag{7}$$

Then from (5) the corresponding mapping can be given as a multivariate Gaussian $\mathcal{P}(s|s_t, a_t) = \mathcal{N}(\mu(s_t, a_t), \Sigma(s_t, a_t))$. Thus from (6) it follows that the target distribution is

$$\mathscr{P}(s|s_t, a_t) = \mathcal{N}(\mu(s_t, a_t), \Sigma(s_t, a_t)) \prod_{i=1}^{\mathcal{D}_s} \frac{1}{1 + \exp(-\kappa(d_i^+ - s^i))} \frac{1}{1 + \exp(-\kappa(s^i - d_i^-))}. \tag{8}$$

Choice of potential energy, kinetic energy and mass matrix: For brevity of notation we drop the explicit dependence of $\mathscr{P}(\cdot)$ on (s_t, a_t) and denote the target distribution as $\mathscr{P}(s)$ defined over the Euclidean space $\mathbb{R}^{\mathcal{D}_s}$. As explained in Section 2.2 we choose the potential energy as

$$\begin{aligned} U(s) = -\log(\mathscr{P}(s)) =& \frac{1}{2}(s - \mu)^T \Sigma^{-1}(s - \mu) - \frac{1}{2} \log\left((2\pi)^{D_s} \det(\Sigma) \right) \\ & - \sum_{i=1}^{D_s} \left[\log\left(1 + \exp(-\kappa(d_i^+ - s^i)) \right) + \log\left(1 + \exp(-\kappa(s^i - d_i^-)) \right) \right]. \end{aligned}$$

We consider an Euclidean metric $\mathcal{M}$ that induces the distance between $\tilde{s}, \bar{s}$ as $d(\tilde{s}, \bar{s}) = (\tilde{s} - \bar{s})^T \mathcal{M}(\tilde{s} - \bar{s})$. Then we define $\mathcal{M}_s \in \mathbb{R}^{\mathcal{D}_s \times \mathcal{D}_s}$ as a diagonal scaling matrix and $\mathcal{M}_r \in \mathbb{R}^{\mathcal{D}_s \times \mathcal{D}_s}$ as a rotation matrix in dimension $\mathcal{D}_s$. With this we can define M as $M = \mathcal{M}_r \mathcal{M}_s \mathcal{M} \mathcal{M}_s^T \mathcal{M}_r^T$. Thus, any metric M that defines an Euclidean structure on the target variable space induces an inverse structure $d(\tilde{v}, \bar{v}) = (\tilde{v} - \bar{v})^T M^{-1}(\tilde{v} - \bar{v})$ on the momentum variable space. This generates a natural family of multivariate Guassian distributions such that $\mathscr{P}(v|s) = \mathcal{N}(0, M)$ leading to the kinetic energy $K(v, s) = -\log \mathscr{P}(v|s) = \frac{1}{2} v^T M^{-1} v$ where M^{-1} is the covariance of the target distribution.

3.2 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 (4) we have

$$\widehat{Q}^{t+1}(s_t, a_t) = r(s_t, a_t) + \frac{\gamma}{|\mathcal{H}_t|} \sum_{s \in \mathcal{H}_t} \max_a Q^t(s, a), \tag{9}$$

for any $(s_t, a_t) \in \Omega_t$. Then we recover the complete matrix Q^{t+1} by using the method given in (2). Thus we have

$$\begin{aligned} Q^{t+1} = & \underset{\widetilde{Q}^{t+1} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}}{\arg\min} \quad \|\widetilde{Q}^{t+1}\|_* \\ & \text{subject to } \mathcal{I}_{\Omega_t}\left(\widetilde{Q}^{t+1}\right) = \mathcal{I}_{\Omega_t}\left(\widehat{Q}^{t+1}\right) \end{aligned} \tag{10}$$

Algorithm 1 Hamiltonian Q-Learning

```
**Inputs:** Discount factor $\gamma$; 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 $\Omega_t$
    **Step 2**: **HMC sampling phase** - Sample a set of next states $\mathcal{H}_t$ according to the target distribution defined in (6)
    **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**
    $Q^{t+1} = \underset{\widetilde{Q}^{t+1} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}}{\arg\min} \quad \|\widetilde{Q}^{t+1}\|_*$
        subject to $\mathfrak{J}_{\Omega_t}\left(\widetilde{Q}^{t+1}\right) = \mathfrak{J}_{\Omega_t}\left(\widehat{Q}^{t+1}\right)$
**end for**
```

Similar to the approach used by [13], we approximate the rank of the Q matrix as the minimum number of singular values that are needed to capture 99% of its nuclear norm.

3.3 Convergence, Boundedness and Sampling Complexity

In this section we provide the main theoretical results of this paper. First, we formally 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.

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

where $|| \cdot ||_F$ is the Frobenius norm (which is 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 mini-max lower bound $\Omega\left(\frac{1}{\epsilon^{\mathcal{D}_s + \mathcal{D}_a + 2}}\right)$ proposed in [15]. 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 given as $(\mathcal{T}Q)(s', a') = \sum_{s \in \mathcal{S}} \mathbb{P}(s|s', a') \left(r(s', a') + \gamma \max_a Q(s, a)\right) \ \forall (s', a') \in \mathcal{S} \times \mathcal{A}$ where $\mathcal{T}$ denotes the Bellman operator. Then, under update rule (3), 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 HMC sampling rule given in Step 3 of the Hamiltonian Q-Learning algorithm (Algorithm 1), the Q function converges to the family of ϵ-optimal Q functions.

Theorem 1 (**Convergence of Q function under HMC**). *Let $\mathfrak{T}$ be an optimality operator under HMC given as $(\mathfrak{T}Q)(s', a') = r(s', a') + \frac{\gamma}{|\mathcal{H}|} \sum_{s \in \mathcal{H}} \max_a Q(s, a), \ \forall (s', a') \in \mathcal{S} \times \mathcal{A}$, where $\mathcal{H}$ is a subset of next states sampled using HMC from the target distribution given in (6). Then, under update rule (4) 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 Q-function convergence proof, i.e. convergence under exhaustive sampling, with a key modification that accounts for the error incurred by HMC sampling. We notice that Q-function error under HMC sampling can be upper bounded by the summation of (*i*) Q-function error under exhaustive sampling and (*ii*) the error between empirical average under HMC sampling and expectation under exhaustive sampling. We note that when Q-function is Lipschitz from central limit theorem for HMC sampling we can upper bound the cumulative error induced by the second term using a constant. Please refer the Supplementary Material for a detailed proof of this theorem. □

The next theorem shows that the Q matrix estimated via a suitable matrix completion technique lies in the ϵ-neighborhood of the corresponding Q function obtained via 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 (9)-(10). 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 we notice that 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^*. Proof follows from upper bounding first term using matrix completion boundedness results and second term using Theorem 1. Please refer Supplementary Material for a detailed proof of this theorem. □

Finally we provide guarantees on the sampling complexity of Hamiltonian Q-Learning algorithm.

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*) Here we briefly state the key steps of our proof. Let T_ϵ be the time step such that learned Q function under Hamiltonian Q-Learning is ϵ optimal. Then number of samples required by Hamiltonian Q-Learning to learn an ϵ optimal Q function can be given as $\sum_{t=1}^{T_\epsilon} |\Omega_t||\mathcal{H}_t|$. 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 next state using HMC samples. Final result follows from combining aforementioned results with convergence and boundedness results obtained in Theorem 1 and 2. A detailed proof of Theorem 3 is given in Supplementary Material. □

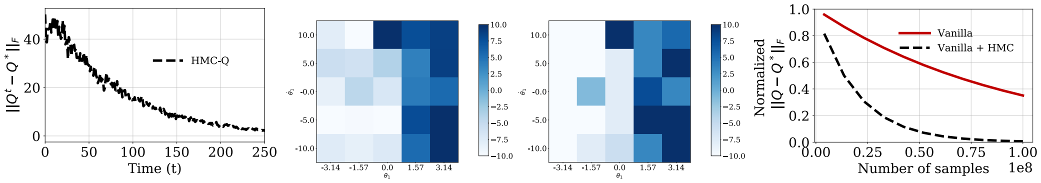
4 Experiments

We illustrate convergence and sample efficiency of Hamiltonian Q-Learning using a high-dimensional system and four benchmark control tasks. Recall that when Q function is Lipschitz convergence in Frobenius norm implies convergence in infinity norm; therefore, we used the Frobenius norm of the difference between the learned Q function and optimal Q^* to illustrate that Hamiltonian Q-Learning converges to at ϵ-optimal Q function.

4.1 Empirical Evaluation for a High-Dimensional System

Experimental setup for a double pendulum on a cart: By letting $x, \dot{x}$ 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 $a \in [-10, 10]$. Then 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 (7) with a Σ which ensures that the range of the state space along direction i approximately equals 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. Please refer Supplementary Material for additional details.

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.,



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

Figure 1: Figure 1(a) illustrates convergence of the Q function learned via 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 exhaustive sampling and Hamiltonian Q-Learning for vanilla Q-Learning.

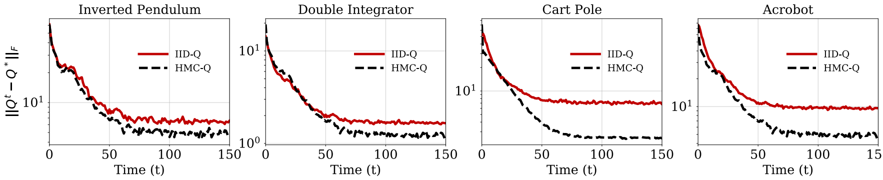


Figure 2: A comparison of convergence of Q function with Hamiltonian Q-Learning and Q-Learning with IID sampling.

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 normalized Frobenius norm of the difference, i.e., Frobenius norm of the difference normalized by its maximum value, between the learned Q function and the optimal Q^* varies with increase in the number of samples. The solid red line shows the accuracy for 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 exhaustive sampling.

4.2 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). Among these four control tasks, the dynamics of inverted pendulum and double integrator evolve on a 2-dimensional state space, whereas cartpole and acrobot are defined on a 4-dimensional state space. We discretize each state space dimension of inverted pendulum and double integrator into 25 distinct values, and each state space dimension of 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. This leads to a Q matrix of size 625×10. Please refer Supplementary Material for additional details about the experimental setup.

Results: Figure 2 shows that 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 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 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 exhaustive sampling and the dashed black lines correspond to Hamiltonian Q-Learning. These

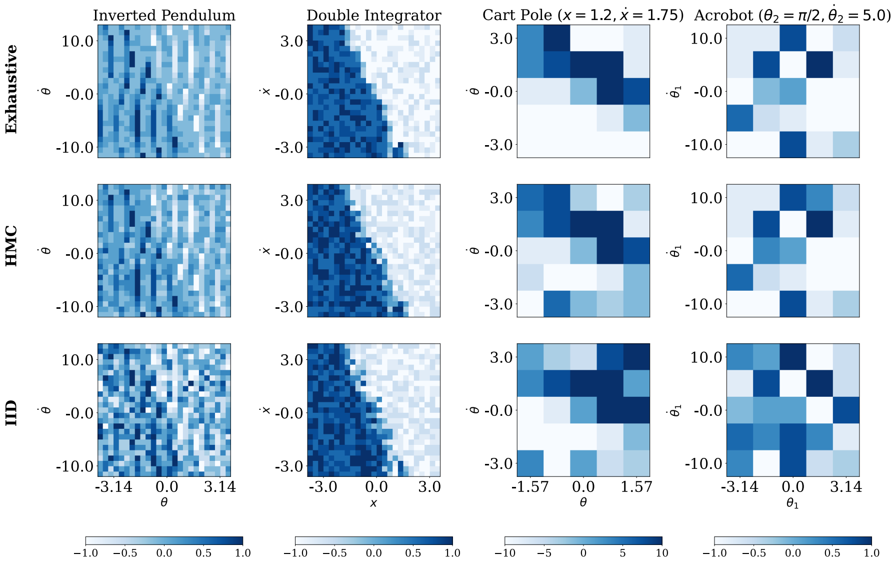


Figure 3: Policy heatmaps for Q-Learning with exhaustive sampling, Hamiltonian Q-Leaning and IID sampling. The color in each cell corresponds to the value of optimal action at the corresponding state.

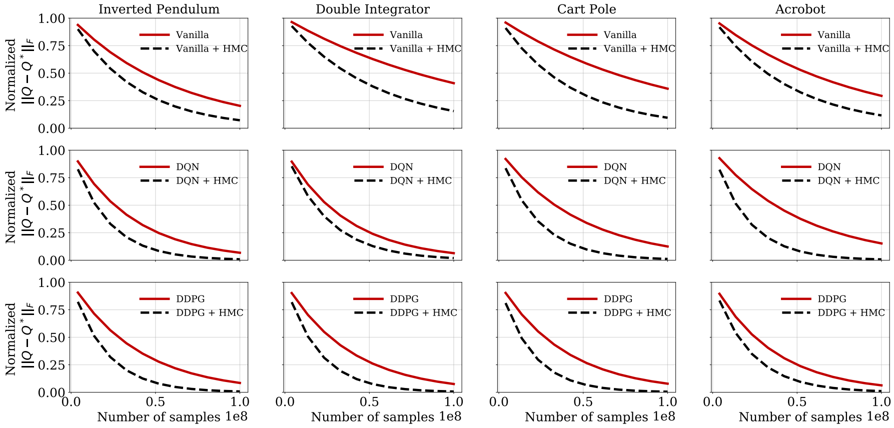


Figure 4: Normalized mean square error, i.e. mean square error divided by it maximum, vs number of samples of Q function with exhaustive sampling and HMC sampling for vanilla Q-Learning, DQN and DDPG. Red solid curve corresponds to HMC sampling and back dotted curve corresponds to HMC sampling.

results show that Hamiltonian Q-Learning can achieve the same level of accuracy with significantly fewer samples.

5 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 obtaining IID samples is impractical. We show, both theoretically and empirically, that the proposed approach can learn accurate estimates of the optimal Q function with much less numbr of samples 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 also. By building upon this aspect, future works will investigate how HMC sampling based methods can improve sample efficiency in multi-agent Q-learning, a system naturally very high-dimensions, 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 (ICML)*, 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] R. M. Neal *et al.*, "MCMC using Hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 2, no. 11, p. 2, 2011.

[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 (ICLR)*, 2020.

[14] D. Shah, D. Song, Z. Xu, and Y. Yang, "Sample efficient reinforcement learning via low-rank matrix estimation," *arXiv:2006.06135*, 2020.

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

[16] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning (ICML)*, 2011, pp. 465–472.

[17] Y. Pan and E. Theodorou, "Probabilistic differential dynamic programming," in *Advances in Neural Information Processing Systems*, 2014, pp. 1907–1915.

[18] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," in *Advances in Neural Information Processing Systems*, 2018, pp. 8224–8234.

[19] R. Dearden, N. Friedman, and S. Russell, "Bayesian q-learning," in *Aaai/iaai*, 1998, pp. 761–768.

[20] A. Koppel, E. Tolstaya, E. Stump, and A. Ribeiro, "Nonparametric stochastic compositional gradient descent for q-learning in continuous markov decision problems," *arXiv preprint arXiv:1804.07323*, 2018.

[21] H. Jeong, C. Zhang, G. J. Pappas, and D. D. Lee, "Assumed density filtering q-learning," *arXiv preprint arXiv:1712.03333*, 2017.

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] S. Y. Lee, C. Sungik, and S.-Y. Chung, "Sample-efficient deep reinforcement learning via episodic backward update," in *Advances in Neural Information Processing Systems*, 2019, pp. 2112–2121.

[29] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[30] F. S. Melo, "Convergence of Q-learning: A simple proof," *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

[31] M. Betancourt, "A conceptual introduction to Hamiltonian Monte Carlo."

[32] M. Betancourt, S. Byrne, S. Livingstone, M. Girolami, *et al.*, "The geometric foundations of Hamiltonian Monte Carlo," *Bernoulli*, vol. 23, no. 4A, pp. 2257–2298, 2017.

[33] K. Neklyudov, M. Welling, E. Egorov, and D. Vetrov, "Involutive MCMC: A Unifying Framework," *arXiv:2006.16653*, 2020.

[34] J. Johns and S. Mahadevan, "Constructing basis functions from directed graphs for value function approximation," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 385–392.

[35] M. Geist and O. Pietquin, "Algorithmic survey of parametric value function approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 6, pp. 845–867, 2013.

[36] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *arXiv:1312.1254*, 2013.

[37] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[38] 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.

[39] K. Yi and F. Doshi-Velez, "Roll-back Hamiltonian Monte Carlo," *arXiv:1709.02855*, 2017.

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

[41] S. Holmes, S. Rubinstein-Salzedo, and C. Seiler, "Curvature and concentration of Hamiltonian Monte Carlo in high dimensions," *arXiv:1407.1114*, 2014.

[42] J. Fan, B. Jiang, and Q. Sun, "Hoeffding's lemma for markov chains and its applications to statistical learning," *arXiv preprint arXiv:1802.00211*, 2018.

[43] D. Maithripala, T. Madhushani, and J. Berg, "A Geometric PID Control Framework for Mechanical Systems," *arXiv:1610.04395*, 2016.

[44] T. Madhushani, D. S. Maithripala, and J. M. Berg, "Feedback regularization and geometric pid control for trajectory tracking of mechanical systems: Hoop robots on an inclined plane," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 3938–3943.

[45] R. McAllister and C. E. Rasmussen, "Data-efficient reinforcement learning in continuous state-action Gaussian-POMDPs," in *Advances in Neural Information Processing Systems*, 2017, pp. 2040–2049.

[46] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning." in *ICLR (Poster)*, 2016. [Online]. Available: http://arxiv.org/abs/1509.02971

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980