Nonparametric Stochastic Compositional Gradient Descent for Q-Learning in Continuous Markov Decision Problems

Alec Koppel*†, Ekaterina Tolstaya**, Ethan Stump†, Alejandro Ribeiro⋆

***Abstract*— We consider Markov Decision Problems defined over continuous state and action spaces, where an autonomous agent seeks to learn a map from its states to actions so as to maximize its long-term discounted accumulation of rewards. We address this problem by considering Bellman's optimality equation defined over action-value functions, which we reformulate into a nested non-convex stochastic optimization problem defined over a Reproducing Kernel Hilbert Space (RKHS). We develop a functional generalization of stochastic quasi-gradient method to solve it, which, owing to the structure of the RKHS, admits a parameterization in terms of scalar weights and past state-action pairs which grows proportionately with the algorithm iteration index. To ameliorate this complexity explosion, we apply Kernel Orthogonal Matching Pursuit to the sequence of kernel weights and dictionaries, which yields a controllable error in the descent direction of the underlying optimization method. We prove that the resulting algorithm, called KQ Learning, converges with probability 1 to a stationary point of this problem, yielding a fixed point of the Bellman optimality operator under the hypothesis that it belongs to the RKHS. Numerical evaluation on the continuous Mountain Car task yields convergent parsimonious learned action-value functions and policies that are competitive with the state of the art.**

I. Introduction

Markov Decision Problems offer a flexible framework to address sequential decision making tasks under uncertainty [1], and have gained broad interest in robotics [2], control [3], finance [4], and artificial intelligence [5]. Despite this surge of interest, few works in reinforcement learning address the computational difficulties associated with continuous state and action spaces in a principled way that guarantees convergence. The goal of this work is to develop new reinforcement learning tools for continuous problems which are provably stable and whose complexity is at-worst moderate.

In the development of stochastic methods for reinforcement learning, one may attempt to estimate the transition density of the Markov Decision Process (MDP) (model-based [6]), perform gradient descent on the value function with respect to the policy (direct policy search [7]), and pursue value function based (model-free [8], [9]) methods which exploit structural properties of the setting to derive fixed point problems called *Bellman equations*. We adopt the latter approach in this work [10], motivated by the fact that an action-value function tells us both how to find a policy and how to evaluate it in terms of the performance metric we have defined, and that a value function encapsulates structural properties of the relationship between states, actions, and rewards.

To understand our proposed approach, consider the fixed point problem defined by Bellman's optimality equation [11]. When the state and action spaces are finite and small enough that expectations are computable, fixed point iterations may be used. When this fails to hold, stochastic fixed point methods, namely, Q-learning [9], may be used, whose convergence may be addressed with asynchronous stochastic approximation theory [12], [13]. This approach is only valid when the action-value (or Q) function may be represented as a matrix. However, when the state and action spaces are infinite, this is no longer true, and the Q-function instead belongs to a generic function space.

In particular, to solve the fixed point problem defined by Bellman's optimality equation when spaces are continuous, one must surmount the fact that it is defined for infinitely many unknowns, one example of Bellman's curse of dimensionality [11]. Efforts to sidestep this issue assume that the Q-function admits a finite parameterization, such as a linear [14], [15] or nonlinear [16] basis expansion, is defined by a neural network [17], or that it belongs to a reproducing kernel Hilbert Space (RKHS) [18], [19]. In this work, we adopt the later nonparametric approach, motivated by the fact that combining fixed point iterations with different parameterizations may cause divergence [20], [21], and in general the Q-function parameterization must be tied to the stochastic update to ensure the convergence of both the function sequence and its parameterization [22].

Our main result is a memory-efficient, nonparametric, stochastic method that converges to a fixed point of the Bellman optimality operator almost surely when it belongs to a RKHS. We obtain this result

* indicates equally contributing authors. This work is supported by grants NSF DGE-1321851 and ARL DCIST CRA W911NF-17-2-0181.

⋆ Department of ESE, University of Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104. Email: {eig, aribeiro}@seas.upenn.edu.

† Computational and Information Sciences Directorate, U.S. Army Research Laboratory, Adelphi, MD, 20783. Email: {alec.e.koppel.civ, ethan.a.stump2.civ}@mail.mil.

by reformulating the Bellman optimality equation as a nested stochastic program (Section II), a topic investigated in operations research [23] and probability [24], [25]. These problems have been addressed in finite settings with stochastic *quasi-gradient* (SQG) methods which use two time-scale stochastic approximation to mitigate the fact that the objective's stochastic gradient is biased with respect to its average [26].

Here, we use a non-parametric generalization of SQG for Q-learning in infinite MDPs (Section III), motivated by its success for policy evaluation in finite [15], [16] and infinite MDPs [27]. However, a function in a RKHS has comparable complexity to the number of training samples processed, which is in general infinite, an issue is often ignored in kernel methods for Markov decision problems [28], [29], [30], [31]. We address this memory bottleneck (the curse of kernelization) by requiring memory efficiency in both the function sample path and in its limit through the use of sparse projections which are constructed greedily via matching pursuit [32], [33], akin to [34], [27]. Greedy compression here is appropriate since (a) kernel matrices induced by arbitrary data streams will likely become ill-conditioned and hence violate assumptions required by convex methods [35], and (b) parsimony is more important than exact recovery as the SQG iterates are not the target signal but rather a noisy stepping stone to Bellman fixed point. Rather than unsupervised forgetting [36], we tie the projection-induced error to guarantee stochastic descent [34], only keeping those dictionary points needed for convergence (Sec. IV).

As a result, we conduct functional SQG descent via sparse projections of the SQG. This maintains a moderate-complexity sample path exactly towards Q^*, which may be made arbitrarily close to a Bellman fixed point by decreasing the regularizer. In contrast to the convex structure in [27], the Bellman optimality equation induces a non-convex cost functional, which requires us to generalize the relationship between SQG for non-convex objectives and coupled supermartingales in [37] to RKHSs. In doing so, we establish that the sparse projected SQG sequence converges almost surely to the Bellman fixed point with decreasing learning rates (Section IV). Moreover, on Continuous Mountain Car [38], we observe that our learned action-value function attains a favorable trade-off between memory efficiency and Bellman error, which then yields a policy whose performance is competitive with the state of the art.

II. Markov Decision Processes

We model an autonomous agent in a continuous space as a Markov Decision Process (MDP) with continuous states $\mathbf{s} \in \mathcal{S} \subset \mathbb{R}^p$ and actions $\mathbf{a} \in \mathcal{A} \subset \mathbb{R}^q$. When in state $\mathbf{s}$ and taking action $\mathbf{a}$, a random transition to state s' occurs according to the conditional probability density $\mathbb{P}(\mathbf{s}'|\mathbf{s},\mathbf{a})$). After the agent to a particular $\mathbf{s}'$ from $\mathbf{s}$, the MDP assigns an instantaneous reward $r(\mathbf{s},\mathbf{a},\mathbf{s}')$, where the reward function is a map $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$.

In Markov Decision problems, the goal is to find the action sequence $\{\mathbf{a}_t\}_{t=0}^{\infty}$ so as to maximize the infinite horizon accumulation of rewards, i.e., the value function: $V(\mathbf{s},\{\mathbf{a}_t\}_{t=0}^{\infty}) := \mathbb{E}_{\mathbf{s}'}[\sum_{t=0}^{\infty}\gamma^t r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t') \mid \mathbf{s}_0 = \mathbf{s}, \{\mathbf{a}_t\}_{t=0}^{\infty}]$. The action-value function $Q(\mathbf{s},\mathbf{a})$ is the conditional mean of the value function given the initial action $\mathbf{a}_0 = \mathbf{a}$:

$$Q(\mathbf{s},\mathbf{a},\{\mathbf{a}_t\}_{t=1}^{\infty}) := \mathbb{E}_{\mathbf{s}'}\left[\sum_{t=0}^{\infty}\gamma^t r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t') \,|\, \mathbf{s}_0=\mathbf{s}, \mathbf{a}_0=\mathbf{a}, \{\mathbf{a}_t\}_{t=1}^{\infty}\right] \tag{1}$$

We consider the case where actions $\mathbf{a}_t$ are chosen according to a stationary stochastic policy, where a policy is a mapping from states to actions: π: $\mathcal{S} \to \mathcal{A}$. We define $Q^*(\mathbf{s},\mathbf{a})$ as the maximum of (1) with respect to the action sequence. The reason for defining action-value functions is that the optimal Q^* may be used to compute the optimal policy π^* as

$$\pi^*(\mathbf{s}) = \operatorname*{argmax}_{\mathbf{a}} Q^*(\mathbf{s},\mathbf{a}) \,. \tag{2}$$

Thus, finding Q^* solves the MDP. Value-function based approaches to MDPs reformulate (2) by shifting the index of the summand in (1) by one, as well as exploiting the time invariance of the Markov transition kernel and the homogeneity of the summand, to derive the Bellman optimality equation:

$$Q^*(\mathbf{s},\mathbf{a}) = \int_{\mathcal{S}} [r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}')]\mathbb{P}(d\mathbf{s}' \mid \mathbf{s},\mathbf{a})) \,. \tag{3}$$

The right-hand side of Equation (3) defines the Bellman optimality operator $\mathscr{B}^*$: $\mathcal{B}(\mathcal{S}\times\mathcal{A}) \to \mathcal{B}(\mathcal{S}\times\mathcal{A})$ over $\mathcal{B}(\mathcal{S}\times\mathcal{A})$, the space of bounded continuous action-value functions Q: $\mathcal{B}(\mathcal{S}\times\mathcal{A}) \to \mathbb{R}$:

$$(\mathscr{B}^*Q)(\mathbf{s},\mathbf{a}) := \int_{\mathcal{S}} [r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}')]\mathbb{P}(d\mathbf{s}' \mid \mathbf{s},\mathbf{a})). \tag{4}$$

[3] [Proposition 5.2] establishes that the fixed point of (4) is the optimal action-value function Q^*. Thus, to solve the MDP, we seek to compute compute the fixed point of (4) for all $(\mathbf{s},\mathbf{a}) \in \mathcal{S}\times\mathcal{A}$.

Compositional Stochastic Optimization Since the above functional fixed point problem is defined for infinitely many unknowns, solving directly in a convergent, moderate complexity manner has eluded researchers for decades. We propose to address this intractability by reformulating it as a nested stochastic optimization problem, inspired by [37], [34], [27]. To do so, we subtract the action value function $Q^*(\mathbf{s},\mathbf{a})$, which

satisfies the optimality equation relation from both sides of (4) and then pull it inside the expectation:

$$0 = \mathbb{E}_{\mathbf{s}'}[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') - Q^*(\mathbf{s},\mathbf{a}) \mid \mathbf{s},\mathbf{a}] \,. \quad (5)$$

Action-value functions that satisfy this condition are equivalent to those which satisfy the quadratic expression for each state-action pair $(\mathbf{s},\mathbf{a})$

$$0 = \frac{1}{2}(\mathbb{E}_{\mathbf{s}'}[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') - Q^*(\mathbf{s},\mathbf{a}) \mid \mathbf{s},\mathbf{a}])^2 \quad (6)$$

We seek to satisfy (5) for all state-action pairs, which is equivalent to the quadratic expression in (6). We seek (6) to be null, independently of any initial state or action, which may be formulated by integrating out the prior of states and actions, yielding the cost functional $L(Q)$:

$$L(Q) \quad (7)$$
$$:= \mathbb{E}_{\mathbf{s},\mathbf{a}}\{\frac{1}{2}(\mathbb{E}_{\mathbf{s}'}[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') - Q(\mathbf{s},\mathbf{a}) \mid \mathbf{s},\mathbf{a}])^2\} \,.$$

Observe that (7) has *nested* expectations, and thus finding the optimal action-value function amounts to solving the compositional stochastic program:

$$Q^* = \operatorname*{argmin}_{Q \in \mathcal{B}(\mathcal{S}\times\mathcal{A})} L(Q) \,. \quad (8)$$

Note that unlike policy evaluation [27], (7) is non-convex. In general, only stationary solutions of (8) may be found. Moreover, (8) is a optimization problem defined over all bounded continuous functions $\mathcal{B}(\mathcal{S}\times\mathcal{A})$, which is impossible to search over. We address this issue next through a hypothesis on the function class.

Reproducing Kernel Hilbert Spaces We propose restricting $\mathcal{B}(\mathcal{S}\times\mathcal{A})$ to be a Hilbert space $\mathcal{H}$ equipped with a unique reproducing kernel, an inner product-like map $\kappa : (\mathcal{S}\times\mathcal{A}) \times (\mathcal{S}\times\mathcal{A}) \to \mathbb{R}$ such that

$$\text{(i)} \langle f, \kappa((\mathbf{s},\mathbf{a}),\cdot)\rangle_{\mathcal{H}} = f(\mathbf{s},\mathbf{a}), \text{(ii)} \mathcal{H} = \overline{\text{span}\{\kappa((\mathbf{s},\mathbf{a}),\cdot)\}} \quad (9)$$

We may apply the Representer Theorem to transform the functional problem into a parametric one [39], [40]. In the Reproducing Kernel Hilbert Space (RKHS), the optimal Q function takes the following form

$$Q(\mathbf{s},\mathbf{a}) = \sum_{n=1}^{N} w_n \kappa((\mathbf{s}_n,\mathbf{a}_n),(\mathbf{s},\mathbf{a})) \quad (10)$$

where (s_n, a_n) is a realization of the random variables in $\mathcal{S}\times\mathcal{A}$. $Q \in \mathcal{H}$ is an expansion of kernel evaluations only at the training samples.

In this work, we restrict the kernel used to be in the family of universal kernels, such as a Gaussian $\kappa((s,a),(s',a')) = \exp\{-\|(s,a)-(s',a')\|_2^2/2c^2\}$, motivated by the fact that a continuous function over a compact set may be approximated uniformly by a function in a RKHS equipped with a universal kernel [41].

To apply the Representer Theorem, we require the cost to be coercive in Q [40], which may be satisfied through use of a Hilbert-norm regularizer, so we define the regularized cost functional $J(Q) = L(Q) + (\lambda/2)\|Q\|_{\mathcal{H}}^2$ and solve the regularized problem (8), i.e.

$$Q^* = \operatorname*{argmin}_{Q\in\mathcal{H}} J(Q) \quad (11)$$

Thus, finding a locally optimal action-value function in an MDP amounts to solving the RKHS-valued compositional stochastic program with a non-convex objective defined by the Bellman optimality equation (4). This action-value function can then be used to obtain the optimal policy (2). In the following section, we turn to iterative stochastic methods to solve (11).

III. Stochastic Quasi-Gradient Method

To solve 11, we propose applying a functional variant of stochastic quasi-gradient (SQG) descent to the loss function $J(Q)$ [cf. (11)]. The reasoning for this approach rather than functional stochastic gradient method is the nested expectations cause the functional stochastic gradient to be biased with respect to its average, and SQG circumvents this issue. Then, we apply the Representer Theorem (10) ("kernel trick") to obtain a tractable parameterization of this optimization sequence, which unfortunately has per-iteration complexity. We then migitate this untenable complexity growth while preserving optimality using greedy compressive methods, inspired by [34], [27].

Now, let's turn to deriving functional SQG. Specifically, to solve (11) using the stochastic quasi-gradient method, we need to compute the functional derivative of the objective $J(Q)$:

$$\nabla_Q J(Q) = \mathbb{E}_{\mathbf{s},\mathbf{a}}\{\nabla_Q \frac{1}{2}(\mathbb{E}_{\mathbf{s}'}[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') - Q(\mathbf{s},\mathbf{a}) \mid \mathbf{s},\mathbf{a}])^2 + \lambda Q \,, \quad (12)$$

where we pull the differential operator inside the expectation. Next, we make use of the chain rule and reproducing property of the kernel. (7) is of the form $J = f \circ g$. We identify $f(u) = \mathbb{E}_{\mathbf{s}\in\mathcal{S},\mathbf{a}\in\mathcal{A}}\{\frac{1}{2}u^2\}$ and $g(Q) = \mathbb{E}_{\mathbf{s}'}[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') - Q(\mathbf{s},\mathbf{a}) \mid \mathbf{s},\mathbf{a}]$.

$$\nabla_Q J(Q) = \mathbb{E}_{\mathbf{s},\mathbf{a}}\{\mathbb{E}_{\mathbf{s}'}\nabla_Q[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') - Q(\mathbf{s},\mathbf{a})] \times \mathbb{E}_{\mathbf{s}'}[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') - Q(\mathbf{s},\mathbf{a})]\} + \lambda Q \quad (13)$$

To compute the directional derivative $\nabla_Q g(Q)$ in (13), we need to define the instantaneous optimal action $\mathbf{a}'$, the instantaneous maximizer of the action-value function.

$$\mathbf{a}' := \operatorname*{argmax}_b Q(\mathbf{s}',\mathbf{b}) \quad (14)$$

Using (14), we define the functional derivative of $g(Q)$ to be $\nabla_Q g(Q) = \mathbb{E}_{\mathbf{s},\mathbf{a}} \gamma\kappa((\mathbf{s}',\mathbf{a}'),\cdot) - \kappa((\mathbf{s},\mathbf{a}),\cdot)$. Finally, the stochastic estimate of (13) can be expressed as

$$\begin{aligned}\hat{\nabla}_Q J(Q,\mathbf{s},\mathbf{a},\mathbf{s}') = &[\gamma\kappa((\mathbf{s}',\mathbf{a}'),\cdot) - \kappa((\mathbf{s},\mathbf{a}),\cdot)] \\ &\times [r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma Q(\mathbf{s}',\mathbf{a}') - Q(\mathbf{s},\mathbf{a})] + \lambda Q\end{aligned} \tag{15}$$

Observe that we cannot obtain unbiased samples of $\nabla_Q J(Q,\mathbf{s},\mathbf{a},\mathbf{s}')$ because the terms inside the inner expectations in (13) are dependent, a problem first identified in [26], [42], [16]. Therefore, we build up the total expectation of one of the terms in (13) while doing stochastic descent with respect to the other. In principle, it is possible to build up the expectation of either term in (13), but the mean of the difference of kernel evaluations is of infinite complexity. On the other hand, the *temporal action difference*, defined as the difference between the action-value function evaluated at state-action pair $(\mathbf{s},\mathbf{a})$ and the action-value function evaluated at next state and the instantaneous maximizing action $(\mathbf{s}',\mathbf{a}')$, i.e.,

$$\delta := r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma Q(\mathbf{s}',\mathbf{a}') - Q(\mathbf{s},\mathbf{a}) \tag{16}$$

is a *scalar*, and thus so is its total expected value. Therefore, for obvious complexity motivations, we build up the total expectation of (16). To do so, we propose recursively averaging realizations of (16) through the following auxiliary sequence z_t, initialized as null $z_0 = 0$:

$$\begin{aligned}\delta_t &:= r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t') + \gamma Q(\mathbf{s}_t',\mathbf{a}_t') - Q(\mathbf{s}_t,\mathbf{a}_t)\,, \\ z_{t+1} &= (1-\beta_t) z_t + \beta_t \delta_t\end{aligned} \tag{17}$$

where $(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')$ is an independent realization of the random triple $(\mathbf{s},\mathbf{a},\mathbf{s}')$ and $\beta_t \in (0,1)$ is a learning rate.

To define the stochastic descent step, we replace the first term inside the outer expectation in (13) with its instantaneous approximation $[\gamma\kappa((\mathbf{s}',\mathbf{a}'),\cdot) - \kappa((\mathbf{s},\mathbf{a}),\cdot)]$ evaluated at a sample triple $(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')$, which yields the stochastic quasi-gradient step:

$$Q_{t+1} = (1-\alpha_t\lambda) Q_t(\cdot) - \alpha_t(\gamma\kappa(\mathbf{s}_t',\mathbf{a}_t',\cdot) - \kappa(\mathbf{s}_t,\mathbf{a}_t,\cdot)) z_{t+1}) \tag{18}$$

where the coefficient $(1-\alpha_t\lambda)$ comes from the regularizer and α_t is a positive scalar learning rate. Moreover, $\mathbf{a}_t' = \operatorname{argmax}_{\mathbf{b}} Q_t(\mathbf{s}',\mathbf{b})$ is the instantaneous Q-function maximizing action. Now, using similar logic to [43], we may extract a computationally tractable parameterization of the infinite dimensional function sequence (18), exploiting properties of the RKHS (9).

Kernel Parametrization Suppose $Q_0 = 0 \in \mathcal{H}$. Then the update in (18) at time t, inductively making use of the Representer Theorem, implies the function Q_t is a kernel expansion of past state-action tuples $(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')$

$$Q_t(s,a) = \sum_{n=1}^{2(t-1)} w_n \kappa(\mathbf{v}_n,(\mathbf{s},\mathbf{a})) = \mathbf{w}_t^T \kappa_{\mathbf{X_t}}((\mathbf{s},\mathbf{a})) \tag{19}$$

The kernel expansion in (19), together with the functional update (18), yields the fact that functional SQG in $\mathcal{H}$ amounts to updating the kernel dictionary $\mathbf{X}_t \in \mathbb{R}^{p\times 2(t-1)}$ and coefficient vector $\mathbf{w}_t \in \mathbb{R}^{2(t-1)}$ as

$$\begin{aligned}\mathbf{X}_{t+1} &= [\mathbf{X}_t, (\mathbf{s}_t,\mathbf{a}_t), (\mathbf{s}_t',\mathbf{a}_t')]\,, \\ \mathbf{w}_{t+1} &= [(1-\alpha_t\lambda)\mathbf{w}_t, \alpha_t z_{t+1}, -\alpha_t\gamma z_{t+1}]\end{aligned} \tag{20}$$

In (20), the coefficient vector $\mathbf{w}_t \in \mathbb{R}^{2(t-1)}$ and dictionary $\mathbf{X}_t \in \mathbb{R}^{p\times 2(t-1)}$ are defined as

$$\begin{aligned}\mathbf{w}_t &= [w_1,\ldots,w_{2(t-1)}]\,, \\ \mathbf{X}_t &= [(\mathbf{s}_1,\mathbf{a}_1),(\mathbf{s}_1',\mathbf{a}_1'),\ldots,(\mathbf{s}_{t-1},\mathbf{a}_{t-1}),(\mathbf{s}_{t-1}',\mathbf{a}_{t-1}')],\end{aligned} \tag{21}$$

and in (19), we introduce the notation $\mathbf{v}_n = (\mathbf{s}_n,\mathbf{a}_n)$ for n even and $\mathbf{v}_n = (\mathbf{s}_n',\mathbf{a}_n')$ for n odd. Moreover, in (19), we make use of a concept called the empirical kernel map associated with dictionary $\mathbf{X}_t$, defined as

$$\begin{aligned}\kappa_{\mathbf{X}_t}(\cdot) = [&(\kappa((\mathbf{s}_1,\mathbf{a}_1),\cdot), \kappa((\mathbf{s}_1',\mathbf{a}_1'),\cdot),\ldots, \\ &\ldots,\kappa((\mathbf{s}_{t-1},\mathbf{a}_{t-1}),\cdot), \kappa((\mathbf{s}_{t-1}',\mathbf{a}_{t-1}'),\cdot)]^T\,.\end{aligned} \tag{22}$$

Observe that (20) causes $\mathbf{X}_{t+1}$ to have two more columns than its predecessor $\mathbf{X}_t$. We define the *model order* as the number of data points (columns) M_t in the dictionary at time t, which for functional stochastic quasi-gradient descent is $M_t = 2(t-1)$. Asymptotically, then, the complexity of storing $Q_t(\cdot)$ is infinite, and even for moderately large training sets is untenable. Next, we address this intractable complexity blowup, inspired by [34], [27], using greedy compression methods [32].

Sparse Stochastic Subspace Projections Since the update step (18) has complexity at least $\mathcal{O}(t)$ due to the parametrization induced by the RKHS, it is impractical in settings with streaming data or arbitrarily large training sets. We address this issue by replacing the stochastic quasi-descent step (18) with an orthogonally projected variant, where the projection is onto a low-dimensional functional subspace of the RKHS $\mathcal{H}_{\mathbf{D}_{t+1}} \subset \mathcal{H}$

$$\begin{aligned}Q_{t+1} = \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}[&(1-\alpha_t\lambda) Q_t(\cdot) \\ &- \alpha_t(\gamma\kappa(\mathbf{s}_t',\mathbf{a}_t',\cdot) - \kappa(\mathbf{s}_t,\mathbf{a}_t,\cdot)) z_{t+1})]\end{aligned} \tag{23}$$

where $\mathcal{H}_{\mathbf{D}_{t+1}} = \operatorname{span}\{((\mathbf{s}_n,\mathbf{a}_n),\cdot)\}_{n=1}^{M_t}$ for some collection of sample instances $\{(\mathbf{s}_n,\mathbf{a}_n)\} \subset \{(\mathbf{s}_t,\mathbf{a}_t)\}_{u\le t}$. We define $\kappa_{\mathbf{D}}(\cdot) = \{\kappa((\mathbf{s}_1,\mathbf{a}_1),\cdot)\ldots\kappa((\mathbf{s}_M,\mathbf{a}_M),\cdot)\}$ and $\kappa_{\mathbf{D},\mathbf{D}}$ as the resulting kernel matrix from this dictionary. We seek function parsimony by selecting dictionaries $\mathbf{D}$ such that $M_t << \mathcal{O}(t)$. Suppose that Q_t is parameterized by model points $\mathbf{D}_t$ and weights $\mathbf{w}_t$. Then, we denote $\tilde{Q}_{t+1}(\cdot) = (1-\alpha_t\lambda) Q_t(\cdot) - \alpha_t(\gamma\kappa(\mathbf{s}_t',\mathbf{a}_t',\cdot) - \kappa(\mathbf{s}_t,\mathbf{a}_t,\cdot)) z_{t+1})$ as the SQG step without projection. This may be represented by dictionary and weight vector [cf. (20)]:

$$\begin{aligned}\tilde{\mathbf{D}}_{t+1} &= [\mathbf{D}_t, (\mathbf{s}_t,\mathbf{a}_t), (\mathbf{s}_t',\mathbf{a}_t')]\,, \\ \tilde{\mathbf{w}}_{t+1} &= [(1-\alpha_t\lambda)\mathbf{w}_t, \alpha_t z_{t+1}, -\alpha_t\gamma z_{t+1}]\,,\end{aligned} \tag{24}$$

where z_{t+1} in (24) is computed by (17) using Q_t obtained from (23):

$$\begin{aligned}\delta_t &:= r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t') + \gamma Q_t(\mathbf{s}_t',\mathbf{a}_t') - Q_t(\mathbf{s}_t,\mathbf{a}_t)\ ,\\ z_{t+1} &= (1-\beta_t)z_t + \beta_t\delta_t\ .\end{aligned} \tag{25}$$

Observe that $\tilde{\mathbf{D}}_{t+1}$ has $\tilde{M}_{t+1} = M_t + 2$ columns which is the length of $\tilde{\mathbf{w}}_{t+1}$. We proceed to describe the construction of the subspaces $\mathcal{H}_{\mathbf{D}_{t+1}}$ onto which the SQG iterates are projected in (23). Specifically, we select the kernel dictionary $\mathbf{D}_{t+1}$ via greedy compression. We form $\mathbf{D}_{t+1}$ by selecting a subset of M_{t+1} columns from $\tilde{\mathbf{D}}_{t+1}$ that best approximates $\tilde{Q}_{t+1}$ in terms of Hilbert norm error. To accomplish this, we use kernel orthogonal matching pursuit [34], [27] with error tolerance ε_t to find a compressed dictionary $\mathbf{D}_{t+1}$ from $\tilde{\mathbf{D}}_{t+1}$, the one that adds the latest samples. For a fixed dictionary $\mathbf{D}_{t+1}$, the update for the kernel weights is a least-squares problem on the coefficient vector:

$$\mathbf{w}_{t+1} = \kappa^{-1}_{\mathbf{D}_{t+1}\mathbf{D}_{t+1}}\kappa_{\mathbf{D}_{t+1}\tilde{\mathbf{D}}_{t+1}}\tilde{\mathbf{w}}_{t+1} \tag{26}$$

We must also tune ε_t to ensure both stochastic descent and finite model order.

We summarize the proposed method, KQ-Learning, in Algorithm 1, the execution of the stochastic projection of the functional SQG iterates onto subspaces $\mathcal{H}_{\mathbf{D}_{t+1}}$. We begin with the initial function null $Q_0 = 0$, with an empty dictionary and coefficients (Step 1). At each step, given an i.i.d. sample $(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')$ and step-size α_t, β_t (Steps 2-5), we compute the unconstrained functional SQG iterate $\tilde{Q}_{t+1}(\cdot) = (1-\alpha_t\lambda)Q_t(\cdot) - \alpha_t(\gamma\kappa(\mathbf{s}_t',\mathbf{a}_t',\cdot) - \kappa(\mathbf{s}_t,\mathbf{a}_t,\cdot))z_{t+1})$ parametrized by $\tilde{\mathbf{D}}_{t+1}$ and $\tilde{\mathbf{w}}_{t+1}$ (Steps 6-7), which are fed into KOMP (Algorithm 2) [34] with budget ε_t, (Step 8).

Algorithm 1 KQ-Learning

Input: $\{\alpha_t,\beta_t,\varepsilon_t\}_{t=0,1,2\ldots}$

1: $Q_0(\cdot) = 0, D_0 = [], w_0 = [], z_0 = 0$
2: **for** $t = 0,1,2,\ldots$ **do**
3: Obtain trajectory $(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')$ via exploratory policy
4: Compute max action: $\mathbf{a}' = \pi_t(\mathbf{s}') = \text{argmax}_{\mathbf{a}'} Q_t(\mathbf{s}_t',\mathbf{a}_t')$
5: Update temporal action diff. δ_t and aux. seq. z_{t+1}

$$\delta_t = r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t') + \gamma Q_t(\mathbf{s}_t',\mathbf{a}_t') - Q_t(\mathbf{s}_t,\mathbf{a}_t)$$

$$z_{t+1} = (1-\beta_t)z_t + \beta_t\delta_t\ .$$

6: Compute functional stochastic quasi-gradient step

$$\tilde{Q}_{t+1} = (1-\alpha_t\lambda)Q_t(\cdot) - \alpha_t(\gamma\kappa(\mathbf{s}_t',\mathbf{a}_t',\cdot) - \kappa(\mathbf{s}_t,\mathbf{a}_t,\cdot))z_{t+1}).$$

7: Update dictionary $\tilde{D}_{t+1} = [D_t,(\mathbf{s},\mathbf{a}),(\mathbf{s}',\mathbf{a}')]$, weights $\tilde{w}_{t+1} = [(1-\alpha_t\lambda)w_t, \alpha_t z_{t+1}, -\alpha_t\gamma z_{t+1}]$.
8: Greedily compress function with KOMP $(Q_{t+1},D_{t+1},w_{t+1}) = \textbf{KOMP}(\tilde{Q}_{t+1},\tilde{D}_{t+1},\tilde{w}_{t+1})$
9: **end for**
10: **return** Q

Algorithm 2 Destructive Kernel Orthogonal Matching Pursuit (KOMP)

Input: function $\tilde{Q}$ defined by dict $\tilde{D} \in \mathbb{R}^{p \times \tilde{M}}$, $\tilde{w} \in \mathbb{R}^{\tilde{M}}$, approx. budget $\varepsilon_t > 0$
Initialize : $Q = \tilde{Q}$, dictionary $D = \tilde{D}$ with indices $\mathcal{I}$, model order $M = \tilde{M}$, coeffs $w = \tilde{w}$.

1: **while** candidate dictionary is non-empty $\mathcal{I} \neq \emptyset$ **do**
2: **for** $j = 1, \ldots, \tilde{M}$ **do**
3: Find minimal approximation error with dictionary element d_j removed

$$\gamma_j = \min_{w_{\mathcal{I}\setminus\{j\}} \in \mathbb{R}^{M-1}} \|\tilde{Q}(\cdot) - \sum_{k \in \mathcal{I}\setminus\{j\}} w_k \kappa(d_k, \cdot)\|_{\mathcal{H}}$$

4: **end for**
5: Find dictionary index minimizing approximation error : $j^* = \operatorname{argmin}_{j \in \mathcal{I}} \gamma_j$
6: **if** minimal approximation error exceeds threshold $\gamma_{j^*} > \varepsilon_t$ **then**
7: **break**
8: **else**
9: Prune dictionary $D \leftarrow D_{\mathcal{I}\setminus\{j^*\}}$
10: Revise set $\mathcal{I} \leftarrow \mathcal{I}\setminus\{j^*\}$ and model order $M \leftarrow M - 1$
11: Compute updated weights w defined by the current dictionary D

$$w = \operatorname*{argmin}_{w \in \mathbb{R}^M} \|\tilde{Q}(\cdot) - w^T \kappa_D(\cdot)\|_{\mathcal{H}}$$

12: **end if**
13: **end while**
14: **return** V, D, w of model order $M \leq \tilde{M}$ such that $\|Q - \tilde{Q}\|_{\mathcal{H}} \leq \varepsilon_t$

In order to implement Algorithm 1, we require the evaluation of the instantaneous maximizing action $\mathbf{a}_t = \operatorname{argmax}_{\mathbf{a}} Q_t((\mathbf{x}, \mathbf{a})$. In the following subsection, we develop a useful numerical procedure to approximately address this task.

Maximizing a Homoscedastic Mixture of Gaussians The KQ-learning algorithm requires an efficient routine for maximizing over the state-action value function $Q(\mathbf{s}, \mathbf{a})$. For a general reproducing kernel κ, maximizing over a weighted sum of kernels is a non-convex optimization problem:

$$\pi(\mathbf{s}) = \operatorname*{argmax}_{\mathbf{a}} \sum_{m=1}^{M} w_m \kappa((\mathbf{s}_m, \mathbf{a}_m), (\mathbf{s}, \mathbf{a})) \qquad (27)$$

We propose an efficient algorithm based on gradient ascent to approximate the maximum and its arguments. In our experiments, we use the Gaussian Radial Basis Function(RBF) kernel, i.e.,

$$\kappa((\mathbf{s}, \mathbf{a}), (\mathbf{s}', \mathbf{a}')) = \exp\{-\frac{1}{2}((\mathbf{s}, \mathbf{a}) - (\mathbf{s}', \mathbf{a}'))\Sigma((\mathbf{s}, \mathbf{a}) - (\mathbf{s}', \mathbf{a}')^T\} \qquad (28)$$

where $\Sigma = \operatorname{diag}(\sigma_1, \cdots, \sigma_{M_t})$, a constant diagonal covariance. We tune the variance of the kernel in each dimension to capture the resolution of the desired policy in both space and action space. For the sum of homoscedastic Gaussian RBFs, we can compute the gradient with respect to an arbitrary action $\mathbf{a}$:

$$(\nabla_{\mathbf{a}} Q)(\mathbf{s}, \mathbf{a}) = Q(\mathbf{s}, \mathbf{a}) \sum_{m=1}^{M} w_m \Sigma_{\mathbf{a}} (\mathbf{a} - \mathbf{a}_m)^T \qquad (29)$$

Since the Q function is non-convex in action vectors $\mathbf{a}$, it is easy to get stuck in undesirable stationary points such as local extrema or saddle points [44]. To reduce the chance that this undesirable outcome transpires, we initialize our gradient ascent iteration at actions defined by model points a_n which maximize the Gaussian mixture model defined by Q:

$$\mathbf{a} = \operatorname*{argmax}_{\mathbf{a}_m} Q(\mathbf{s}, \mathbf{a}_m) \qquad (30)$$

Next, we use gradient ascent to refine our estimate of the global maximum of Q for state $\mathbf{s}$.

$$\mathbf{a} \leftarrow \mathbf{a} + \gamma Q(\mathbf{s}, \mathbf{a}) \sum_{m=1}^{M} w_m \Sigma_{\mathbf{a}} (\mathbf{a} - \mathbf{a}_m)^T \qquad (31)$$

We summarize the proposed algorithm in Algorithm 3: First, given state $\mathbf{s}$, find the maximizing action $\mathbf{a}_m$ among the dictionary elements (Step 1). Next, perform gradient ascent on $Q(\mathbf{s}, \mathbf{a})$ with respect to $\mathbf{a}$, given $\mathbf{s}$ (Steps 2-4).

Algorithm 3 Argmax over Sum of RBFs

Input: function Q defined by dict $D = [(\mathbf{s}_1, \mathbf{a}_1), \ldots, (\mathbf{s}_M, \mathbf{a}_M)] \in \mathbb{R}^{p \times M}$, $w \in \mathbb{R}^M$, state $\mathbf{s}$, step size γ, precision ε

1: Find the maximum Q function value among the dictionary elements projected onto the subspace such that $\mathbf{s}_m = \mathbf{s}$. Break ties randomly.

$$\mathbf{a} = \operatorname*{argmax}_{\mathbf{a}_m} Q(\mathbf{s}, \mathbf{a}_m)$$

2: **while** $\gamma \|\Delta \mathbf{a}\|_2 > \varepsilon$ **do**
3: Compute the constrained gradient $\Delta \mathbf{a} = \nabla_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a})$, where $\Sigma_{\mathbf{a}}$ is the diagonal covariance for the actions

$$\Delta \mathbf{a} = Q(\mathbf{s}, \mathbf{a}) \sum_{m=1}^{M} w_m \Sigma_{\mathbf{a}} (\mathbf{a} - \mathbf{a}_m)^T$$

4: $\mathbf{a} \leftarrow \mathbf{a} + \gamma \Delta \mathbf{a}$
5: **end while**
6: **return** argument of the maximizer, $\mathbf{a}$

IV. Convergence Analysis

In this section, we shift focus to the task of establishing that the sequence of action-value function estimates generated by Algorithm 1 actually yield a locally optimal solution to the Bellman optimality equation, which, given intrinsic the non-convexity of the problem setting, is the best one may hope for in general through use of numerical stochastic optimization methods. Our analysis extends the ideas of coupled supermartingales in reproducing kernel Hilbert spaces [27], which have been used to establish convergent policy evaluation approaches in infinite MDPs (a convex problem), to non-convex settings, and further generalizes the non-convex vector-valued setting of [37].

Before proceeding with the details of the technical setting, we introduce a few definitions which simplify derivations greatly. In particular, for further reference, we use (14) to define $\mathbf{a}'_t = \operatorname{argmax}_{\mathbf{a}'_t} Q_t(\mathbf{s}'_t, \mathbf{a}'_t)$, the instantaneous maximizer of the action-value function and defines the direction of the gradient. We also define the functional stochastic quasi-gradient of the regularized objective

$$\hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) = (\gamma\kappa((\mathbf{s}'_t, \mathbf{a}'_t), \cdot) - \kappa((\mathbf{s}_t, \mathbf{a}_t), \cdot))z_{t+1} + \lambda Q_t \tag{32}$$

and its sparse-subspace projected variant as

$$\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) = (Q_t - \mathcal{P}_{\mathcal{H}_{D_{t+1}}}[Q_t - \alpha_t \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)])/\alpha_t \tag{33}$$

Note that the update may be rewritten as a stochastic projected quasi-gradient step rather than a stochastic quasi-gradient step followed by a set projection, i.e.,

$$Q_{t+1} = Q_t - \alpha_t \tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) \tag{34}$$

With these definitions, we may state our main assumptions required to establish convergence of Algorithm 1.

Assumption 1 *The state space $\mathcal{S} \subset \mathbb{R}^p$ and action space $\mathcal{A} \subset \mathbb{R}^q$ are compact, and the reproducing kernel map may be bounded as*

$$\sup_{\mathbf{s}\in\mathcal{S}, \mathbf{a}\in\mathcal{A}} \sqrt{\kappa((\mathbf{s}, \mathbf{a}), (\mathbf{s}, \mathbf{a}))} = K < \infty \tag{35}$$

Assumption 2 *The temporal action difference δ and auxiliary sequence z satisfy the zero-mean, finite conditional variance, and Lipschitz continuity conditions, respectively,*

$$\mathbb{E}[\delta \,|\, \mathbf{s}, \mathbf{a}] = \bar{\delta}\ ,\ \mathbb{E}[(\delta - \bar{\delta})^2] \leq \sigma_\delta^2\ ,\ \mathbb{E}[z^2 \,|\, \mathbf{s}, \mathbf{a}] \leq G_\delta^2 \tag{36}$$

where σ_δ and G_δ are positive scalars, and $\bar{\delta} = \mathbb{E}\{\delta \mid \mathbf{s}, \mathbf{a}\}$ is defined as the expected value of the temporal action difference conditioned on the state $\mathbf{s}$ and action $\mathbf{a}$.

Assumption 3 *The functional gradient of the temporal action difference is an unbiased estimate for $\nabla_Q J(Q)$ and the difference of the reproducing kernels expression has finite conditional variance:*

$$\mathbb{E}[(\gamma\kappa((\mathbf{s}'_t, \mathbf{a}'_t), \cdot) - \kappa((\mathbf{s}_t, \mathbf{a}_t), \cdot))\delta] = \nabla_Q J(Q) \tag{37}$$

$$\mathbb{E}\{\|\gamma\kappa((\mathbf{s}'_t, \mathbf{a}'_t), \cdot) - \kappa((\mathbf{s}_t, \mathbf{a}_t), \cdot)\|_{\mathcal{H}}^2 | \mathcal{F}_t\} \leq G_Q^2 \tag{38}$$

Moreover, the projected stochastic quasi-gradient of the objective has finite second conditional moment as

$$\mathbb{E}\{\|\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)\|_{\mathcal{H}}^2 | \mathcal{F}_t\} \leq \sigma_Q^2 \tag{39}$$

and the temporal action difference is Lipschitz continuous with respect to the action-value function Q. Moreover, for any two distinct δ and $\bar{\delta}$, we have

$$\|\delta - \bar{\delta}\| \leq L_Q \|Q - \tilde{Q}\|_{\mathcal{H}} \tag{40}$$

with Q, $\tilde{Q} \in \mathcal{H}$ distinct Q-functions; $L_Q > 0$ is a scalar.

Assumption 1 regarding the compactness of the state and action spaces of the MDP holds for most application settings and limits the radius of the set from which the MDP trajectory is sampled. The mean and variance properties of the temporal difference stated in Assumption 2 are necessary to bound the error in the descent direction associated with the stochastic sub-sampling and are required to establish convergence of stochastic methods. Assumption 3 is similar to Assumption 2, but instead of establishing bounds on the stochastic approximation error of the temporal difference, limits stochastic error variance in the RKHS. Moreover, (40) is justified since the maximum of a continuous function is Lipschitz in the infinity norm, which can be related to the Hilbert norm through a constant factor. These are natural extensions of the conditions needed for vector-valued stochastic compositional gradient methods.

The compactness of $\mathcal{S}$ and $\mathcal{A}$ (Assumption 1) implies that $\mathcal{H}$ is a compact function space, which together with the closedness of Hilbert subspaces $\mathcal{H}_{D_t}$, mean Q_t is contained within compact sets for all t due to the use of set projections in (23), meaning

$$\|Q_t\|_{\mathcal{H}} \leq D \text{ for all } t\ , \tag{41}$$

where $D > 0$ is some positive constant.

Next, we turn to establishing some technical results which are necessary precursors to the proofs of the main stability results.

Proposition 1 *Given independent identical realizations $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)$ of the random triple $(\mathbf{s}, \mathbf{a}, \mathbf{s}')$, the difference between the projected stochastic functional quasi-gradient and the stochastic functional quasi-gradient of the instantaneous cost is bounded for all t as*

$$\|\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) - \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)\|_{\mathcal{H}} \leq \frac{\varepsilon_t}{\alpha_t} \tag{42}$$

Where $\alpha_t > 0$ denotes the algorithm step size and $\varepsilon_t > 0$ is the compression budget parameter of the KOMP algorithm.

Proof: As in Proposition 1 of [27], Consider the square-Hilbert norm difference of $\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)$ and $\hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)$ defined by (32) and (33)

$$\begin{aligned}\|\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) - \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)\|_{\mathcal{H}} = \\ \|(Q_t - \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}[Q_t - \alpha_t \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)])/\alpha_t \\ - \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)\|^2_{\mathcal{H}} \end{aligned} \tag{43}$$

Multiply and divide $\hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)$ by α_t and reorder terms to write

$$\begin{aligned}&\| \frac{(Q_t - \alpha_t \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t))}{\alpha_t} \\ &\quad - \frac{(\mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}[Q_t - \alpha_t \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)])}{\alpha_t} \|^2_{\mathcal{H}} \\ &= \frac{1}{\alpha_t^2} \|(Q_t - \alpha_t \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)) \\ &\quad - (\mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}[Q_t - \alpha_t \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)])\|^2_{\mathcal{H}} \\ &= \frac{1}{\alpha_t^2} \|\tilde{Q}_{t+1} - Q_{t+1}\|^2_{\mathcal{H}} \leq \frac{\varepsilon_t^2}{\alpha_t^2} \end{aligned} \tag{44}$$

where we have pulled the nonnegative scalar α_t outside of the norm on the second line and substituted the definition of $\tilde{Q}_{t+1}$ and Q_{t+1}. We also apply the KOMP residual stopping criterion from Algorithm 2, $\|\tilde{Q}_{t+1} - Q_{t+1}\| \leq \varepsilon_t$ to yield (42). ■

Lemma 1 *Denote the filtration $\mathcal{F}_t$ as the time-dependent sigma-algebra containing the algorithm history $(\{Q_u, z_u\}_{u=0}^t \cup \{\mathbf{s}_u, \mathbf{a}_u, \mathbf{s}'_u\}_{u=0}^{t-1}) \subset \mathcal{F}_t$. Let Assumptions 1-3 hold true and consider the sequence of iterates defined by Algorithm 1. Then:*

i. The conditional expectation of the Hilbert-norm difference of action-value functions at the next and current iteration satisfies the relationship

$$\mathbb{E}[\|Q_{t+1} - Q_t\|^2_{\mathcal{H}} | \mathcal{F}_t] \leq 2\alpha_t^2 (G_\delta^2 G_Q^2 + \lambda D^2) + 2\varepsilon_t^2 \tag{45}$$

ii. The auxiliary sequence z_t with respect to the conditional expectation of the temporal action difference $\bar{\delta}_t$ (defined in Assumption 2) satisfies

$$\begin{aligned}\mathbb{E}[(z_{t+1} - \bar{\delta}_t)^2 \,|\, \mathcal{F}_t] \leq (1 - \beta_t)(z_t - \bar{\delta}_{t-1})^2 \\ + \frac{L_Q}{\beta_t} \|Q_t - Q_{t-1}\|^2_{\mathcal{H}} + 2\beta_t^2 \sigma_\delta^2 \end{aligned} \tag{46}$$

iii. Algorithm 1 generates a sequence of Q-functions that satisfy the stochastic descent property with respect to the Bellman error $J(Q)$ [cf. (11)]:

$$\begin{aligned}\mathbb{E}[J(Q_{t+1}) \,|\, \mathcal{F}_t] \leq J(Q_t) - \alpha_t \left(1 - \frac{\alpha_t G_Q^2}{\beta_t}\right) \|\nabla_Q J(Q)\|^2 \\ + \frac{\beta_t}{2} \mathbb{E}[(\bar{\delta}_t - z_{t+1})^2 \,|\, \mathcal{F}_t] + \frac{L_Q \sigma_Q^2 \alpha_t^2}{2} \\ + \varepsilon_t \|\nabla_Q J(Q_t)\|_{\mathcal{H}} \,, \end{aligned} \tag{47}$$

Proof : *Lemma 1*(i) Consider the Hilbert-norm difference of action-value functions at the next and current iteration and use the definition of Q_{t+1}

$$\|Q_{t+1} - Q_t\|^2_{\mathcal{H}} = \alpha_t^2 \|\tilde{\nabla}_Q J(Q_t, z_{t+1}; (\mathbf{s}_t, \mathbf{a}_t), (\mathbf{s}'_t, \mathbf{a}'_t)\|^2_{\mathcal{H}} \tag{48}$$

We add and subtract the functional stochastic quasi-gradient $\hat{\nabla}_Q J(Q_t, z_{t+1}; (\mathbf{s}_t, \mathbf{a}_t), (\mathbf{s}'_t, \mathbf{a}'_t)$ from (48) and apply the triangle inequality $(a+b)^2 \leq 2a^2 + 2b^2$ which holds for any $a, b > 0$.

$$\begin{aligned}\|Q_{t+1} - Q_t\|^2_{\mathcal{H}} \leq 2\alpha_t^2 \|\hat{\nabla}_Q J(Q_t, z_{t+1}; (\mathbf{s}_t, \mathbf{a}_t), (\mathbf{s}'_t, \mathbf{a}'_t)\|^2_{\mathcal{H}} \\ + 2\alpha_t^2 \|\hat{\nabla}_Q J(Q_t, z_{t+1}; (\mathbf{s}_t, \mathbf{a}_t), (\mathbf{s}'_t, \mathbf{a}'_t) \\ - \tilde{\nabla}_Q J(Q_t, z_{t+1}; (\mathbf{s}_t, \mathbf{a}_t), (\mathbf{s}'_t, \mathbf{a}'_t)\|^2_{\mathcal{H}} \end{aligned} \tag{49}$$

Now, we may apply Proposition 1 to the second term. Doing so and computing the expectation conditional on the filtration $\mathcal{F}_t$ yields

$$\begin{aligned}&\mathbb{E}[\|Q_{t+1} - Q_t\|^2_{\mathcal{H}}] \mid \mathcal{F}] \\ &= 2\alpha_t^2 \mathbb{E}[\|\hat{\nabla}_Q J(Q_t, z_{t+1}; (\mathbf{s}_t, \mathbf{a}_t), (\mathbf{s}'_t, \mathbf{a}'_t)\|^2_{\mathcal{H}} \mid \mathcal{F}_t] + 2\varepsilon_t^2 \end{aligned} \tag{50}$$

Using the Cauchy-Schwarz inequality together with the Law of Total Expectation and the definition of the functional stochastic quasi-gradient to upper estimate the first term on the right-hand side of (50) as

$$\begin{aligned}&\mathbb{E}[\|Q_{t+1} - Q_t\|^2_{\mathcal{H}} \mid \mathcal{F}_t] \\ &\leq 2\alpha_t^2 \mathbb{E}\{\|\gamma \kappa((\mathbf{s}'_t, \mathbf{a}'_t), \cdot) - \kappa((\mathbf{s}_t, \mathbf{a}_t), \cdot)\|^2_{\mathcal{H}} \\ &\quad \times \mathbb{E}[z_{t+1}^2 \mid \mathbf{s}_t, a_t] \mid \mathcal{F}_t\} + 2\alpha_t^2 \lambda \|Q_t\|^2_{\mathcal{H}} + 2\varepsilon_t^2 \end{aligned} \tag{51}$$

Now, use the fact that z_{t+1} has a finite second conditional moment [cf. (36)], yielding

$$\begin{aligned}&\mathbb{E}[\|Q_{t+1} - Q_t\|^2_{\mathcal{H}} \mid \mathcal{F}_t] \\ &\leq 2\alpha_t^2 G_\delta^2 \mathbb{E}[\|\gamma \kappa((\mathbf{s}'_t, \mathbf{a}'_t), \cdot) - \kappa((\mathbf{s}_t, \mathbf{a}_t), \cdot)\|^2_{\mathcal{H}} \mid \mathcal{F}_t] \\ &\quad + 2\alpha_t^2 \lambda \|Q_t\|^2_{\mathcal{H}} + 2\varepsilon_t^2 \end{aligned} \tag{52}$$

From here, we may use the fact that the functional gradient of the temporal action-difference $\gamma \kappa((\mathbf{s}'_t, \mathbf{a}'_t), \cdot) - \kappa((\mathbf{s}_t, \mathbf{a}_t), \cdot)$ has a finite second conditional moment (36) and that the Q function sequence is bounded (41) to write:

$$\mathbb{E}[\|Q_{t+1} - Q_t\|^2_{\mathcal{H}} \mid \mathcal{F}_t] \leq 2\alpha_t^2 (G_\delta^2 G_V^2 + \lambda^2 D^2) + 2\varepsilon_t^2 \tag{53}$$

which is as stated in Lemma 1(i). ■

Proof : *Lemma 1*(ii) Begin by defining the scalar quantity e_t as the difference of mean temporal-action

differences scaled by the forgetting factor β_t, i.e. $e_t = (1-\beta_t)(\bar{\delta}_t - \bar{\delta}_{t-1})$. Then, we consider the difference of the evolution of the auxiliary variable z_{t+1} with respect to the conditional mean temporal action difference $\bar{\delta}_t$, plus the difference of the mean temporal differences:

$$\begin{aligned} z_{t+1} - \bar{\delta}_t + e_t =& (1-\beta_t)z_t + \beta_t \delta_t - [(1-\beta_t)\bar{\delta}_t + \beta_t \bar{\delta}_t] \\ &+ (1-\beta_t)(\bar{\delta}_t - \bar{\delta}_{t-1}) \end{aligned} \tag{54}$$

where we make use of the definition of z_{t+1}, the fact that $\bar{\delta}_t = \{(1-\beta_t)\bar{\delta}_t + \beta_t \bar{\delta}_t\}$ and the definition of e_t on the right-hand side of (54). Observe that the result then simplifies to $z_{t+1} - \bar{\delta}_t + e_t = (1-\beta_t)z_t + \beta_t(\bar{\delta}_t - \bar{\delta}_{t-1})$ by grouping like terms and canceling the redundant $\bar{\delta}_t$. Squaring (54), using this simplification, yields

$$\begin{aligned} &(z_{t+1} - \bar{\delta}_t + e_t)^2 \\ &\quad = (1-\beta_t)^2 (z_t - \bar{\delta}_{t-1})^2 + \beta_t^2 (\delta_t - \bar{\delta}_t)^2 \\ &\qquad + 2(1-\beta_t)\beta_t (z_t - \bar{\delta}_{t-1})(\delta_t - \bar{\delta}_t) \end{aligned} \tag{55}$$

Now, let's compute the expectation conditioned on the algorithm history $\mathcal{F}_t$ to write

$$\begin{aligned} &\mathbb{E}[(z_{t+1} - \bar{\delta}_t + e_t)^2 \mid \mathcal{F}_t] \\ &\quad = (1-\beta_t)^2 (z_t - \bar{\delta}_{t-1})^2 + \beta_t^2 \mathbb{E}[(\delta_t - \bar{\delta}_t)^2 \mid \mathcal{F}_t] \\ &\qquad + 2(1-\beta_t)\beta_t (z_t - \bar{\delta}_{t-1}) \mathbb{E}[(\delta_t - \bar{\delta}_t) \mid \mathcal{F}_t] \end{aligned} \tag{56}$$

We apply the assumption that the temporal action difference δ_t is an unbiased estimator for its conditional mean $\bar{\delta}_t$ with finite variance (Assumption 2) to write

$$\mathbb{E}[(z_{t+1} - \bar{\delta}_t + e_t) \big| \mathcal{F}_t] = (1-\beta_t)^2 (z_t - \bar{\delta}_{t-1})^2 + \beta_t^2 \sigma_\delta^2 \tag{57}$$

We obtain an upper estimate on the conditional mean square of $z_{t+1} - \bar{\delta}_t$ by using the inequality $\|a+b\|^2 \leq (1+\rho)\|a\|^2 + (1+1\rho)\|b\|^2$ which holds for any $\rho > 0$: set $a = z_{t+1} - \bar{\delta}_t + e_t$, $b = -e_t$, $\rho = \beta_t$ to write

$$(z_{t+1} - \bar{\delta}_t)^2 \leq (1+\beta_t)(z_{t+1} - \bar{\delta}_t + e_t)^2 + \left(1 + \frac{1}{\beta_t}\right) e_t^2 \tag{58}$$

Observe that (58) provides an upper-estimate of the square sub-optimality $(z_{t+1} - \bar{\delta}_t)^2$ in terms of the squared error sequence $(z_{t+1} - \bar{\delta}_t + e_t)^2$. Therefore, we can compute the expectation of (58) conditional on $\mathcal{F}_t$ and substitute (57) for the terms involving the error sequence $(z_{t+1} - \bar{\delta}_t + e_t)^2$, which results in gaining a factor of $(1+\beta_t)$ on the right-hand side. Collecting terms then yields

$$\begin{aligned} &\mathbb{E}[(z_{t+1} - \bar{\delta}_t)^2 \mid \mathcal{F}_t] \\ &= (1+\beta_t)[(1-\beta_t)^2 (z_t - \bar{\delta}_{t-1})^2 + \beta_t^2 \sigma_\delta^2] + \left(\frac{1+\beta_t}{\beta_t}\right) e_t^2 \end{aligned} \tag{59}$$

Using the fact that $(1-\beta_t^2)(1-\beta_t) \leq (1-\beta_t)$ for the first term and $(1-\beta_t)\beta_t^2 \leq 2\beta_t^2$ for the second to simplify

$$\begin{aligned} \mathbb{E}[(z_{t+1} - \bar{\delta}_t)^2 \mid \mathcal{F}_t] =& (1-\beta_t)(z_t - \bar{\delta}_{t-1})^2 + 2\beta_t^2 \sigma_\delta^2 \\ &+ \left(\frac{1+\beta_t}{\beta_t}\right) e_t^2 \end{aligned} \tag{60}$$

We can bound the term involving e_t, which represents the difference of mean temporal differences. By definition, we have $|e_t| = (1-\beta_t)|(\bar{\delta}_t - \bar{\delta}_{t-1})|$:

$$(1-\beta_t)|(\bar{\delta}_t - \bar{\delta}_{t-1})| \leq (1-\beta_t) L_Q \|Q_t - Q_{t-1}\|_{\mathcal{H}} , \tag{61}$$

where we apply the Lipschitz continuity of the temporal difference with respect to the action-value function [cf. (40)]. Substitute the right-hand side of (61) and simplify the expression in the last term as $(1-\beta_t^2)/\beta_t \leq 1/\beta_t$ to conclude (47). ■

Proof: *Lemma 1*(iii) Following the proof of Theorem 4 of [37], we begin by considering the Taylor expansion of $J(Q)$ and applying the fact that it has Lipschitz continuous functional gradients to upper-bound the second-order terms. Doing so yields the quadratic upper bound:

$$\begin{aligned} J(Q_{t+1}) \leq J(Q_t) &+ \langle \nabla J(Q_t), Q_{t+1} - Q_t \rangle_{\mathcal{H}} \\ &+ \frac{L_Q}{2} \|Q_{t+1} - Q_t\|_{\mathcal{H}}^2 . \end{aligned} \tag{62}$$

Substitute the fact that the difference between consecutive action-value functions is the projected quasi-stochastic gradient $Q_{t+1} - Q_t = -\alpha_t \tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_t')$ (34) into (62) .

$$\begin{aligned} J(Q_{t+1}) \leq& J(Q_t) - \alpha_t \langle \nabla J(Q_t), \tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_t') \rangle_{\mathcal{H}} \\ &+ \frac{L_Q \alpha_t^2}{2} \|\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_t')\|_{\mathcal{H}}^2 . \end{aligned} \tag{63}$$

Subsequently, we use the short-hand notation $\hat{\nabla}_Q J(Q_t) := \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_t')$ and $\tilde{\nabla}_Q J(Q_t) := \tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_t')$ for the stochastic and projected stochastic quasi-gradients, (32) and (33), respectively. Now add and subtract the inner-product of the functional gradient of J with the stochastic gradient, scaled by the step-size $\alpha_t \langle \nabla J(Q_t), \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_t') \rangle_{\mathcal{H}}$, as well as $\alpha_t \|\nabla_Q J(Q_t)\|^2$ into above expression and gather like terms.

$$\begin{aligned} J(Q_{t+1}) \leq& J(Q_t) - \alpha_t \|\nabla_Q J(Q)\|^2 + \frac{L_Q \alpha_t^2}{2} \|\tilde{\nabla}_Q J(Q_t)\|_{\mathcal{H}}^2 \\ &- \alpha_t \langle \nabla J(Q_t), \tilde{\nabla}_Q J(Q_t) - \hat{\nabla}_Q J(Q_t) \rangle_{\mathcal{H}} \\ &+ \alpha_t \langle \nabla J(Q_t), \nabla J(Q_t) - \hat{\nabla}_Q J(Q_t) \rangle_{\mathcal{H}} \end{aligned} \tag{64}$$

Observe that the last two terms on the right-hand side of (64) are terms associated with the directional error between the true gradient and the stochastic quasi-gradient, as well as the stochastic quasi-gradient with respect to the projected stochastic quasi-gradient. The former term may be addressed through the error bound derived from

the KOMP stopping criterion in Proposition 1, whereas the later may be analyzed through judicious use of the Law of Total Expectation and Assumptions 2 - 3.

Let's proceed to address the second term on the right-hand side of (64). Begin by applying Cauchy-Schwarz to write

$$\begin{aligned}|-\alpha_t\langle\nabla J(Q_t),\tilde{\nabla}_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\rangle_{\mathcal{H}}| \\ \leq \alpha_t\|\nabla J(Q_t)\|_{\mathcal{H}}\|\tilde{\nabla}_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\|_{\mathcal{H}} \end{aligned} \tag{65}$$

Now, apply Proposition 1 to $\|\tilde{\nabla}_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\|_{\mathcal{H}}$, the Hilbert-norm error induced by sparse projections on the right-hand side of (65) and cancel the redundant factor of α_t:

$$\alpha_t\langle\nabla J(Q_t),\tilde{\nabla}_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\rangle_{\mathcal{H}} \leq \varepsilon_t\|\nabla J(Q_t)\|_{\mathcal{H}} \tag{66}$$

Next, let's address the last term on the right-hand side of (64). To do so, we will exploit Assumptions 2 - 3 and the Law of Total Expectation. First, consider the expectation of this term, ignoring the multiplicative step-size factor, while applying (37):

$$\begin{aligned}&\mathbb{E}\big[\nabla J(Q_t),\nabla_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\rangle_{\mathcal{H}}\,\big|\,\mathcal{F}_t\big] \\ &=\Big\langle\nabla_Q J(Q_t),\mathbb{E}[(\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot))(\bar{\delta}_t-z_{t+1})\big|\mathcal{F}_t]\Big\rangle_{\mathcal{H}}\end{aligned} \tag{67}$$

In (67), we pull the expectation inside the inner-product, using the fact that $\nabla_Q J(Q)$ is deterministic. Note on the right-hand side of (67), by using (37), we have $\bar{\delta}_t$ inside the expectation in the above expression rather than a realization δ_t. Now, apply Cauchy-Schwartz to the above expression to obtain

$$\begin{aligned}&\Big\langle\nabla_Q J(Q_t),\mathbb{E}[(\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot))(\bar{\delta}_t-z_{t+1})\big|\mathcal{F}_t]\Big\rangle_{\mathcal{H}} \\ &\leq\|\nabla_Q J(Q_t)\|_{\mathcal{H}}\mathbb{E}[\|(\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot))\|_{\mathcal{H}} \\ &\quad\times|\bar{\delta}_t-z_{t+1}|\,\big|\,\mathcal{F}_t]\end{aligned} \tag{68}$$

From here, apply the inequality $ab\leq\frac{\rho}{2}a^2+\frac{1}{2\rho}b^2$ for $\rho>0$ with $a=|\bar{\delta}_t-z_{t+1}|$, and $b=\alpha_t\|\nabla_Q J(Q_t)\|\|\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot)\|_{\mathcal{H}}$, and $\rho=\beta_t$ to the preceding expression:

$$\begin{aligned}&\|\nabla_Q J(Q_t)\|_{\mathcal{H}}\mathbb{E}[\|(\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot))\|_{\mathcal{H}}|\bar{\delta}_t-z_{t+1}|\big|\mathcal{F}_t]\Big\rangle_{\mathcal{H}} \\ &\leq\frac{\beta_t}{2}\mathbb{E}[(\bar{\delta}_t-z_{t+1})^2\mid\mathcal{F}_t] \\ &+\frac{\alpha_t^2}{2\beta_t}\|\nabla J(Q_t)\|_{\mathcal{H}}^2\mathbb{E}[\|\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot)\|_{\mathcal{H}}^2\,\big|\,\mathcal{F}_t]\end{aligned} \tag{69}$$

To (69), let's apply Assumption 3 regarding the finite second conditional of the difference of reproducing kernel maps (38) to the second term, which when substituted into the right-hand side of the expectation of (64) conditional on $\mathcal{F}_t$, yields

$$\begin{aligned}\mathbb{E}[J(Q_{t+1})\,|\,\mathcal{F}_t]\leq J(Q_t)-\alpha_t\left(1-\frac{\alpha_t G_Q^2}{\beta_t}\right)\|\nabla_Q J(Q)\|^2 \\ +\frac{\beta_t}{2}\mathbb{E}[(\bar{\delta}_t-z_{t+1})^2\mid\mathcal{F}_t]+\frac{L_Q\sigma_Q^2\alpha_t^2}{2} \\ +\varepsilon_t\|\nabla_Q J(Q_t)\|_{\mathcal{H}}\,,\end{aligned} \tag{70}$$

where we have also applied the fact that the projected stochastic quasi-gradient has finite conditional variance (39) and gathered like terms to conclude (47).

■

Lemma 1 is may be seen as a nonparametric extension of Lemma 2 and A.1 of [37], or an extension of Lemma 6 in [27] to the non-convex case. Now, we may use Lemma 1 to connect the function sequence generated by Algorithm 1 to a special type of stochastic process called a coupled supermartingale, and therefore prove that Q_t converges to a stationary point of the Bellman error with probability 1. To the best of our knowledge, this is a one of a kind result.

Lemma 2 *(Coupled Supermartingale Theorem) [45], [37]. Let $\{\xi_t t\}$, $\{\zeta_t\}$, $\{u_t\}$, $\{\bar{u}_t\}$, $\{\eta_t\}$, $\{\theta_t\}$, $\{\varepsilon_t\}$, $\{\mu_t\}$, $\{\nu_t\}$ be sequences of nonnegative random variables such that*

$$\begin{aligned}\mathbb{E}\{\xi_{t+1}\mid G_t\}\leq(1+\eta_t)\xi_t-u_t+c\theta_t\zeta_t+\mu_t\,, \\ \mathbb{E}\{\zeta_{t+1}\mid G_t\}\leq(1-\theta_t)\zeta_t-\bar{u}_t+\varepsilon_t\xi_t+\nu_t\end{aligned} \tag{71}$$

where $G_t=\{\xi_s,\xi_s,u_s,\bar{u}_s,\eta_s,\theta_s,\varepsilon_s,\mu_s,\nu_s\}_{s=0}^t$ is the filtration and $c>0$ is a scalar. Suppose the following summability conditions hold almost surely:

$$\sum_{t=0}^{\infty}\eta_t<\infty\,,\sum_{t=0}^{\infty}\varepsilon_t<\infty\,,\sum_{t=0}^{\infty}\mu_t<\infty\,,\sum_{t=0}^{\infty}\nu_t<\infty \tag{72}$$

Then ξ_t and ζ_t converge almost surely to two respective nonnegative random variables, and we may conclude that almost surely

$$\sum_{t=0}^{\infty}u_t<\infty\,,\sum_{t=0}^{\infty}\bar{u}_t<\infty\,,\sum_{t=0}^{\infty}\theta_t\zeta_t<\infty \tag{73}$$

Theorem 1 *Consider the sequence z_t and $\{Q_t\}$ as stated in Algorithm 1. Assume the regularizer is positive $\lambda>0$, Assumptions 1-3 hold, and the step-size conditions hold, with $C>0$ a positive constant:*

$$\sum_{t=1}^{\infty}\alpha_t=\infty,\sum_{t=1}^{\infty}\beta_t=\infty,\sum_{t=1}^{\infty}\alpha_t^2+\beta_t^2+\frac{\alpha_t^2}{\beta_t}<\infty,\varepsilon_t=C\alpha_t^2 \tag{74}$$

Then $\|\nabla_Q J(Q)\|_{\mathcal{H}}$ converges to null with probability 1, and hence Q_t attains a stationary point of (11)*. In particular, the limit of Q_t achieves the regularized Bellman fixed point restricted to the RKHS.*

Proof : *Theorem 1* We use the relations established in Lemma 1 to construct a coupled supermatingale of the form 2. First, we use Lemma 1(ii)(46) to provide an upper bound on Lemma 1(iii) (47).

$$\mathbb{E}[J(Q_{t+1})\,|\,\mathcal{F}_t] \leq J(Q_t) - \alpha_t\left(1-\frac{\alpha_t G_Q^2}{\beta_t}\right)\|\nabla_Q J(Q_t)\|^2 + \frac{\beta_t}{2}((1-\beta_t)(z_t-\bar{\delta}_{t-1})^2 + \frac{L_Q}{\beta_t}\|Q_t - Q_{t-1}\|_{\mathcal{H}}^2 + 2\beta_t^2\sigma_\delta^2) + \frac{L_Q\sigma_Q^2\alpha_t^2}{2} + \varepsilon_t\|\nabla_Q J(Q_t)\|_{\mathcal{H}} \quad (75)$$

We introduce three restrictions on the learning rate, expectation rate, and parsimony constant in order to simplify (75). First, we assume that $\beta_t \in (0,1)$ for all t. Next, we choose $\varepsilon_t = \alpha_t^2$. Lastly, we restrict $1-\frac{\alpha_t G_Q^2}{\beta_t} > 0$, which results in the condition: $\frac{\alpha_t}{\beta_t} < \frac{1}{G_Q^2}$. Then, we simplify and group terms of (75).

$$\mathbb{E}[J(Q_{t+1})\,|\,\mathcal{F}_t] \leq J(Q_t) - \alpha_t\|\nabla_Q J(Q_t)\|^2 + \frac{\beta_t}{2}(z_t-\bar{\delta}_{t-1})^2 + \frac{L_Q}{2}\|Q_t-Q_{t-1}\|_{\mathcal{H}}^2 + \beta_t^3\sigma_\delta^2 + \alpha_t^2\left(\frac{L_Q\sigma_Q^2}{2} + \|\nabla_Q J(Q_t)\|_{\mathcal{H}}\right) \quad (76)$$

Next, we aim to connect the result of (75) to the form of Lemma 2 via the identifications:

$$\xi_t = J(Q_t)\,,\zeta_t = (z_t-\bar{\delta}_{t-1})^2\,,\theta_t=\beta_t\,,c=1/2 \quad (77)$$
$$u_t = \alpha_t\|\nabla_Q J(Q_t)\|^2, \eta_t = 0$$
$$\mu_t = \frac{L_Q}{2}\|Q_t-Q_{t-1}\|_{\mathcal{H}}^2 + \beta_t^3\sigma_\delta^2 + \alpha_t^2\left(\frac{L_Q\sigma_Q^2}{2} + \|\nabla_Q J(Q_t)\|_{\mathcal{H}}\right)$$

Observe that $\sum \mu_t < \infty$ due to the upper bound on $\|Q_t - Q_{t-1}\|_{\mathcal{H}}^2$ provided by Lemma 1(45) and the summability conditions for α_t^2 and β_t^2 (74).

Next, we turn our attention to identifying terms in Lemma 1 (ii) (46) according to Lemma 2 in addition to (77).

$$\nu_t = \frac{L_Q}{\beta_t}\|Q_t-Q_{t-1}\|_{\mathcal{H}}^2 + 2\beta_t^2\sigma_\delta^2\,,\varepsilon_t = 0\,,\bar{u}_t = 0 \quad (78)$$

The summability of ν_t can be shown as follows: the expression $\|Q_t - Q_{t-1}\|_{\mathcal{H}}^2/\beta_t$ which is order $\mathcal{O}(\alpha_t^2/\beta_t)$ in conditional expectation by Lemma 1(i). Sum the resulting conditional expectation for all t, which by the summability of the sequence $\sum_t \alpha_t^2/\beta_t < \infty$ is finite. Therefore, $\sum_t \|Q_t - Q_{t-1}\|_{\mathcal{H}}^2/\beta_t < \infty$ almost surely. We also require $\sum_t \beta_t^2 < \infty$ (74) for the summability of the second term of (78)

Together with the conditions on the step-size sequences α_t and β_t, the summability conditions of Lemma 2 are satisfied, which allows to conclude that $\xi_t = J(Q_t)$ and $\zeta_t = (z_t - \bar{\delta}_{t-1})^2$ converge to two non-negative random variables w.p. 1, and that

$$\sum_t \alpha_t\|\nabla_Q J(Q_t)\|^2 < \infty, \quad \sum_t \beta_t(z_t-\bar{\delta}_{t-1})^2 < \infty \quad (79)$$

almost surely. Then, the summability of u_t taken together with non-summability of α_t and β_t (74) indicates that the limit infimum of the norm of the gradient of the cost goes to null.

$$\liminf_{t\to\infty}\|\nabla_Q J(Q_t)\| = 0\,,\liminf_{t\to\infty}(z_t-\bar{\delta}_{t-1})^2 = 0 \quad (80)$$

almost surely. From here, given $\liminf_{t\to\infty}\|\nabla_Q J(Q_t)\|_{\mathcal{H}} = 0$, we can apply almost the exact same argument by contradiction as [37] to conclude that the whole sequence $\|\nabla_Q J(Q_t)\|_{\mathcal{H}}$ converges to null with probability 1, which is repeated here for completeness.

Consider some $\eta > 0$ and observe that $\|\nabla_Q J(Q_t)\|_{\mathcal{H}} \leq \eta$ for infinitely many t. Otherwise, there exists t_0 such that $\sum_{t=0}^{\infty}\|\alpha_t\nabla_Q J(Q_t)\|_{\mathcal{H}}^2 \geq \sum_{t=t_0}^{\infty}\alpha_t\eta^2 = \infty$ which contradicts (79). Therefore, there exists a closed set $\bar{\mathcal{H}} \subset \mathcal{H}$ such that $\{Q_t\}$ visits $\bar{\mathcal{H}}$ infinitely often, and

$$\|\nabla_Q J(Q)\|_{\mathcal{H}} \begin{cases} \leq \eta \text{ for } Q \in \bar{\mathcal{H}} \\ > \eta \text{ for } Q \notin \bar{\mathcal{H}}, Q \in \{Q_t\} \end{cases} \quad (81)$$

Suppose to the contrary that there exists a limit point $\tilde{Q}$ such that $\|\nabla_Q J(\tilde{Q})\|_{\mathcal{H}} > 2\eta$. Then there exists a closed set $\tilde{\mathcal{H}}$, i.e., a union of neighborhoods of all Q_t's such that $\|\nabla_Q J(Q_t)\|_{\mathcal{H}} > 2\eta$, with $\{Q_t\}$ visiting $\tilde{\mathcal{H}}$ infinitely often, and

$$\|\nabla_Q J(Q)\|_{\mathcal{H}} \begin{cases} \geq 2\eta \text{ for } Q \in \tilde{\mathcal{H}} \\ < 2\eta \text{ for } Q \notin \tilde{\mathcal{H}}, Q \in \{Q_t\} \end{cases} \quad (82)$$

Using the continuity of ∇J and $\eta > 0$, we have that $\bar{\mathcal{H}}$ and $\tilde{\mathcal{H}}$ are disjoint: $\text{dist}(\bar{\mathcal{H}}, \tilde{\mathcal{H}}) > 0$. Since $\{Q_t\}$ enters both $\bar{\mathcal{H}}$ and $\tilde{\mathcal{H}}$ infinitely often, there exists a subsequence $\{Q_t\}_{t\in\mathcal{T}} = \{\{Q_t\}_{t=k_i}^{j_i-1}\}$ (with $\mathcal{T} \subset \mathbb{Z}^+$) that enters $\bar{\mathcal{H}}$ and $\tilde{\mathcal{H}}$ infinitely often, with $Q_{k_i} \in \bar{\mathcal{H}}$ and $Q_{j_i} \in \tilde{\mathcal{H}}$ for all i. Therefore, for all i, we have

$$\|\nabla_Q J(Q_{k_i})\|_{\mathcal{H}} \geq 2\eta > \|\nabla_Q J(Q_t)\|_{\mathcal{H}} > \eta \geq \|\nabla_Q J(Q_{j_i})\|_{\mathcal{H}} \quad \text{for } t = k_i+1,\ldots,j_i-1 \quad (83)$$

Therefore, we can write

$$\sum_{t\in\mathcal{T}}\|Q_{t+1}-Q_t\|_{\mathcal{H}} = \sum_{i=1}^{\infty}\sum_{t=k_i}^{j_i-1}\|Q_{t+1}-Q_t\|_{\mathcal{H}} \geq \sum_{i=1}^{\infty}\|Q_{k_i}-Q_{j_i}\|_{\mathcal{H}} \geq \text{dist}(\bar{\mathcal{H}},\tilde{\mathcal{H}}) = \infty \quad (84)$$

However, we may also write that

$$\infty > \sum_{t=0}^{\infty}\alpha_t\|\nabla J(Q_t)\|_{\mathcal{H}}^2 \geq \sum_{t\in\mathcal{T}}\alpha_t\|\nabla J(Q_t)\|_{\mathcal{H}}^2 > \eta^2\sum_{t\in\mathcal{T}}\alpha_t \quad (85)$$

Then, using the fact that the sets $\mathcal{X}$ and $\mathcal{A}$ are compact, there exist some $M > 0$ such that $\|Q_{t+1} - Q_t\|_{\mathcal{H}} \leq \alpha_t \|\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)\|_{\mathcal{H}} \leq M\alpha_t$ for all t, using the fact that $\varepsilon_t = \alpha_t^2$. Therefore,

$$\sum_{t \in \mathcal{T}} \|Q_{t+1} - Q_t\|_{\mathcal{H}} \leq M \sum_{t \in \mathcal{T}} \alpha_t < \infty \tag{86}$$

which contradicts (86). Therefore, there does not exist any limit point $\tilde{Q}$ such that $\|\nabla_Q J(\tilde{Q})\|_{\mathcal{H}} > 2\eta$. By making η arbitrarily small, it means that there does not exist any limit point that is nonstationary. Moreover, we note that the set of such sample paths occurs with probability 1, since the preceding analysis applies to all sample paths which satisfy (79). Thus, any limit point of Q_t is a stationary point of $J(Q)$ almost surely. ■

Theorem 1 establishes that Algorithm 1 converges almost surely to a stationary solution of the problem (11) defined by the Bellman optimality equation in a continuous MDP. This is one of the first Lyapunov stability results for Q-learning in continuous state-action spaces with nonlinear function parameterizations, which are intrinsically necessary when the Q-function does not admit a lookup table (matrix) representation, and should form the foundation for value-function based reinforcement learning in continuous spaces. A key feature of this result is that the complexity of the function parameterization will not grow untenably large due to the use of our KOMP-based compression method which ties the sparsification bias ε_t to the algorithm step-size α_t. In the following section, we investigate the empirical validity of the proposed approach on a classic autonomous control task called the Continuous Mountain Car.

V. Experiments

We benchmark KQ-Learning (Algorithm 1) on a classic control problem, the Continuous Mountain Car [40], which is featured in OpenAI Gym [38]. In this problem, the state space is $p = 2$ dimensional, consisting of position and velocity, bounded within $[-1.2, 0.6]$ and $[-0.07, 0.07]$, respectively. The action space is $q = 1$ dimensional: force on the car, within the interval $[-1, 1]$. The reward function is 100 when the car reaches the goal at position 0.6, and $-0.1a^2$ for any action a.

We used Gaussian RBFs with a fixed non-isotropic kernel bandwidth $\sigma_1 = 0.8$, $\sigma_2 = 0.07$, $\sigma_3 = 1.0$ for all experiments. The relevant parameters are the step-sizes α and β, the regularizer λ, and the approximation error constant, C, where we fix the compression budget $\varepsilon = C\alpha^2$. These learning parameters were tuned through a grid search procedure, which yielded the following selections: $\gamma = 0.99$, $\sigma = [0.8, 0.07, 1.0]$, $\lambda = 10^{-6}$, $\varepsilon = 0.1$, $\mathrm{T} = 5 \times 10^5$, $\alpha = 0.5$, $\beta = 0.5$. As the agent traverses the environment, we select actions randomly, initially with probability 1, and then exponentially decay this likelihood to 0.1 after 10^5 exploratory training steps.

The results of this experiment are given in Figure 1: here we plot the normalized Bellman test error Fig. 1b, defined by the sample average approximation of (7) divided by the Hilbert norm of Q_t over a collection of generated test trajectories, as well as the average rewards during training (Fig. 1a), and the model order, i.e., the number of training examples in the kernel dictionary (Fig. 1c), all relative to the number of training samples processed. Observe that the Bellman test error converges and the interval average rewards approach 90, which is the benchmark used to designate a policy as "solving" Continuous Mountain Car. This is comparable to existing top entries on the OpenAI Leaderboard [38], such as Deep Deterministic Policy Gradient [46] . Moreover, we obtain this result with a complexity reduction by orders of magnitude relative to existing methods for Q-function and policy representation.

Additionally, few heuristics are required to ensure KQ-Learning converges, which is in contrast to neural network approaches to Q-learning. One shortcoming of our implementation is its sample efficiency, which could improved through an experience replay buffer. Such methods re-reveal past trajectory data to the agent based on the magnitude of their temporal action difference, for example, and have been shown to accelerate learning. Alternative, variance reduction, acceleration, or Quasi-Newton methods would improve the learning rate.

An additional feature of our method is the interpretability of the resulting Q function, which we use to plot the value function (2a) and policy (2b). One key metric is the coverage of the kernel points in the state-action space. We can make conclusions about the generalizability of the policy by the density of the model points throughout the space. Also, we can interpret which past experiences make an impact on the current value and policy evaluation. This may have particular importance in mechanical or econometric applications, where the model points represent physical phenomena or specific events in financial markets.

References

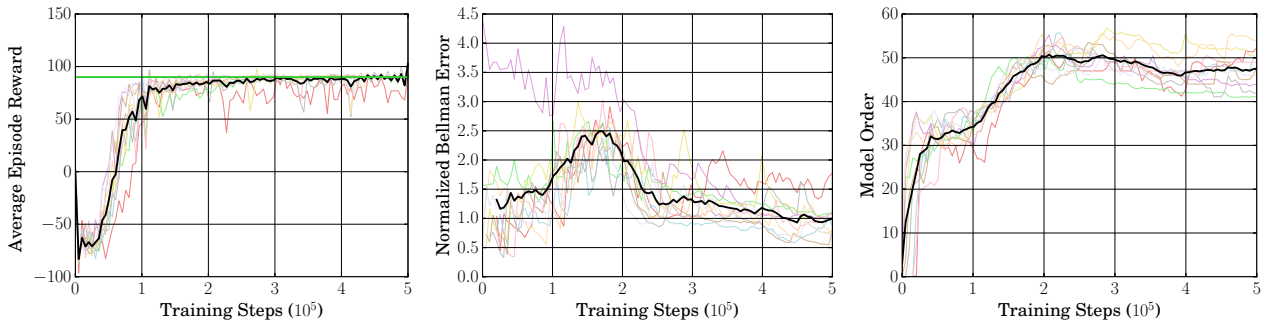
[1] R. Bellman, "The theory of dynamic programming," DTIC Document, Tech. Rep., 1954.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, p. 0278364913495721, 2013.

[3] D. P. Bertsekas and S. E. Shreve, *Stochastic optimal control: The discrete time case*. Academic Press, 1978, vol. 23.

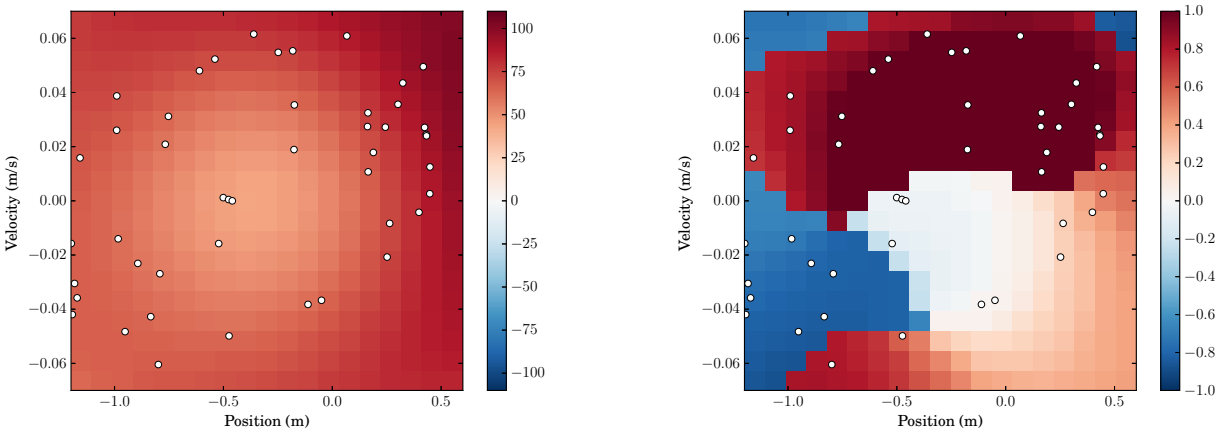
[4] M. Rásonyi, L. Stettner *et al.*, "On utility maximization in discrete-time financial market models," *The Annals of Applied Probability*, vol. 15, no. 2, pp. 1367–1395, 2005.

[5] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.



(a) Average Training Reward (b) Normalized Test Bellman Error (c) Model Order of Q

Fig. 1: Results of 10 experiments over 500,000 training steps were averaged (black curve) to demonstrate the learning progress for the effective, convergent, and parsimonious solution. Fig. 1a shows the average reward obtained by the ε-greedy policy during training. An average reward over 90 (green) indicates that we have solved Continuous Mountain Car, steering towards the goal location. Fig. 1b shows the Bellman error for testing samples (7) normalized by the Hilbert norm of Q, which converges to a small non-zero value. Fig. 1c shows the number of points parameterizing the kernel dictionary of Q during training, which remains under 55 on average. Overall, we solve Continuous Mountain Car with a complexity reduction by orders of magnitude relative to existing methods [17], [46].



(a) Value function $V(\mathbf{s})$ derived from limiting $Q(\mathbf{s},\mathbf{a})$ (b) Policy $\pi(\mathbf{s})$ derived from limiting $Q(\mathbf{s},\mathbf{a})$

Fig. 2: The learned Q-function is easily interpretable: we may visualize the value function, $V(s) = \max_a Q(s,a)$ (2a) and corresponding policy $\pi(s) = \text{argmax}_a Q(s,a)$ (2b). In Fig. 2a, the color indicates the value of the state, which is highest (dark red) near the goal 0.6. At this position, for any velocity, the agent receives an award of 100 and concludes the episode. In Fig. 2b, the color indicates the force on the car (action), for a given position and velocity (state). The learned policy takes advantage of the structure of the environment to accelerate the car without excess force inputs. The dictionary points are pictured in white and provide coverage of the state-action space.

[6] K. Mitkovska-Trendova, R. Minovski, and D. Boshkovski, "Methodology for transition probabilities determination in a markov decision processes model for quality-accuracy management," *Journal of Engineering Management and Competitiveness (JEMC)*, vol. 4, no. 2, pp. 59–67, 2014.

[7] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[8] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

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

[10] W. B. Powell and J. Ma, "A review of stochastic algorithms with continuous value function approximation and some new approximate policy iteration algorithms for multidimensional continuous applications," *Journal of Control Theory and Applications*, vol. 9, no. 3, pp. 336–352, 2011.

[11] R. Bellman, *Dynamic Programming*, 1st ed. Princeton, NJ, USA: Princeton University Press, 1957.

[12] J. N. Tsitsiklis, "Asynchronous stochastic approximation and q-learning," *Machine Learning*, vol. 16, no. 3, pp. 185–202, 1994.

[13] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural computation*, vol. 6, no. 6, pp. 1185–1201, 1994.

[14] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *Proceedings of the 25th international conference on Machine learning*.

ACM, 2008, pp. 664–671.

[15] R. S. Sutton, H. R. Maei, and C. Szepesvári, "A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation," in *Advances in neural information processing systems*, 2009, pp. 1609–1616.

[16] S. Bhatnagar, D. Precup, D. Silver, R. S. Sutton, H. R. Maei, and C. Szepesvári, "Convergent temporal-difference learning with arbitrary smooth function approximation," in *Advances in Neural Information Processing Systems*, 2009, pp. 1204–1212.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[18] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *Journal of mathematical analysis and applications*, vol. 33, no. 1, pp. 82–95, 1971.

[19] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *International Conference on Computational Learning Theory*. Springer, 2001, pp. 416–426.

[20] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *In Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 30–37.

[21] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE transactions on automatic control*, vol. 42, no. 5, pp. 674–690, 1997.

[22] N. K. Jong and P. Stone, "Model-based function approximation in reinforcement learning," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 2007, p. 95.

[23] A. Shapiro, D. Dentcheva *et al.*, *Lectures on stochastic programming: modeling and theory*. Siam, 2014, vol. 16.

[24] A. Korostelev, "Stochastic recurrent procedures: Local properties," *Nauka: Moscow (in Russian)*, 1984.

[25] V. R. Konda and J. N. Tsitsiklis, "Convergence rate of linear two-time-scale stochastic approximation," *Annals of applied probability*, pp. 796–819, 2004.

[26] Y. Ermoliev, "Stochastic quasigradient methods and their application to system optimization," *Stochastics: An International Journal of Probability and Stochastic Processes*, vol. 9, no. 1-2, pp. 1–36, 1983.

[27] A. Koppel, G. Warnell, E. Stump, P. Stone, and A. Ribeiro, "Breaking bellman's curse of dimensionality: Efficient kernel gradient temporal difference," *arXiv preprint arXiv:1709.04221 (Submitted to AAAI 2017)*, 2017.

[28] D. Ormoneit and Ś. Sen, "Kernel-based reinforcement learning," *Machine learning*, vol. 49, no. 2-3, pp. 161–178, 2002.

[29] S. Grünewälder, G. Lever, L. Baldassarre, M. Pontil, and A. Gretton, "Modelling transition dynamics in mdps with rkhs embeddings," in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, vol. 1, 2012, pp. 535–542.

[30] A.-m. Farahmand, C. Ghavamzadeh, Mohammadand Szepesvári, and S. Mannor, "Regularized policy iteration with nonparametric function spaces," *Journal of Machine Learning Research*, vol. 17, no. 139, pp. 1–66, 2016.

[31] B. Dai, N. He, Y. Pan, B. Boots, and L. Song, "Learning from conditional distributions via dual kernel embeddings," *arXiv preprint arXiv:1607.04579*, 2016.

[32] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[33] G. Lever, J. Shawe-Taylor, R. Stafford, and C. Szepesvari, "Compressed conditional mean embeddings for model-based reinforcement learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[34] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious Online Learning with Kernels via Sparse Projections in Function Space," *ArXiv e-prints*, Dec. 2016.

[35] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[36] Y. Engel, S. Mannor, and R. Meir, "Bayes meets bellman: The gaussian process approach to temporal difference learning," in *Proc. of the 20th International Conference on Machine Learning*, 2003.

[37] M. Wang, E. X. Fang, and H. Liu, "Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions," *Mathematical Programming*, vol. 161, no. 1-2, pp. 419–449, 2017.

[38] Openai gym - continuous mountain car.

[39] V. Norkin and M. Keyzer, "On stochastic optimization and statistical learning in reproducing kernel hilbert spaces by support vector machines (svm)," *Informatica*, vol. 20, no. 2, pp. 273–292, 2009.

[40] A. Argyriou, C. A. Micchelli, and M. Pontil, "When is there a representer theorem? vector versus matrix regularizers," *Journal of Machine Learning Research*, vol. 10, no. Nov, pp. 2507–2529, 2009.

[41] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, no. Dec, pp. 2651–2667, 2006.

[42] V. S. Borkar and S. P. Meyn, "The ode method for convergence of stochastic approximation and reinforcement learning," *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 447–469, 2000.

[43] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," in *Advances in neural information processing systems*, 2002, pp. 785–792.

[44] M. Carreira-Perpinan and C. Williams, "On the number of modes of a gaussian mixture," in *Scale Space Methods in Computer Vision*. Springer, 2003, pp. 625–640.

[45] D. P. Bertsekas and S. Shreve, *Stochastic optimal control: the discrete-time case*, 2004.

[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," *arXiv preprint arXiv:1509.02971*, 2015.