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. Under constant learning rates, we further obtain convergence to a small Bellman error that depends on the chosen learning rates. Numerical evaluation on the Continuous Mountain Car and Inverted Pendulum tasks yields convergent parsimonious learned action-value functions, policies that are competitive with the state of the art, and exhibit reliable, reproducible learning behavior.**

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.

It is well-known that approaches featuring the "deadly triad" [5] of function approximation, bootstrapping (e.g. temporal-difference learning), and off-policy training are in danger of divergence, and the most well-understood techniques for ensuring convergence in a stochastic gradient descent context are those based on Gradient Temporal Difference (GTD) [11]. Though the final algorithm looks similar, our approach could be considered as an alternative formulation and analysis of the GTD family of algorithms centered on a flexible RKHS representation that lets us address problems with nonlinear, continuous state and action spaces in a natural way.

To understand our proposed approach, consider the fixed point problem defined by Bellman's optimality equation [12]. 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 [13], [14]. 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 [12]. Efforts to sidestep this issue assume that the Q-function admits a finite parameterization, such as a linear [15], [11] 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, non-parametric, 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 by reformulating the Bellman optimality equation as a nested stochastic program (Section II), a topic investigated in operations research [23] and probability

* indicates equally contributing authors. This work is supported by NSF DGE-1321851, ARL DCIST CRA W911NF-17-2-0181, Intel DevCloud and Intel Science and Technology Center for Wireless Autonomous Systems (ISTC-WAS).

⋆ 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.

[24], [25]. These problems have been addressed in finite settings with stochastic *quasi-gradient* (SQG) methods [26] which use two time-scale stochastic approximation to mitigate the fact that the objective's stochastic gradient not available due to its dependence on a *second expectation*, which is referred to as the double sampling problem in [11].

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 [11], [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 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 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 stepping stone to Bellman fixed point. Rather than unsupervised forgetting [36], we tie the projection-induced error to guarantee stochastic descent [34], only keeping dictionary points needed for convergence.

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 (Theorem 1) to the Bellman fixed point with decreasing learning rates (Section IV) and to a small Bellman error whose magnitude depends on the learning rates when learning rates are held constant (Theorem 2). Use of constant learning rates allows us to further guarantee that the memory of the learned Q function remains under control. Moreover, on Continuous Mountain Car [38] and the Inverted Pendulum [39], 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 in terms of episode average reward accumulation.

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 transitions 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 $\mathbf{a}_0 = \mathbf{a}$:

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

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

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

where a policy is a map from states to actions: π: $\mathcal{S} \to \mathcal{A}$. 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, use the time invariance of the Markov transition kernel, and the homogeneity of the summand, to derive the Bellman optimality equation:

$$Q^*(\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'}\Big[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q^*(\mathbf{s}',\mathbf{a}') \,\Big|\, \mathbf{s},\mathbf{a}\Big]. \quad (3)$$

where the expectation is taken with respect to the conditional distribution $\mathbb{P}(d\mathbf{s}' \mid \mathbf{s},\mathbf{a})$ of the state $\mathbf{s}'$ given the state action pair $(\mathbf{s},\mathbf{a})$. 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: $\mathcal{B}(\mathcal{S}\times\mathcal{A}) \to \mathbb{R}$:

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

[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 the fixed point of (4) for all $(\mathbf{s},\mathbf{a}) \in \mathcal{S}\times\mathcal{A}$.

Compositional Stochastic Optimization. The functional fixed point equation in (3) has to be simultaneously satisfied for all state action pairs $(\mathbf{s},\mathbf{a})$. Alternatively, we can integrate (3) over an arbitrary distribution that is dense around any pair $(\mathbf{s},\mathbf{a})$ to write a nested stochastic optimization problem [37], [34], [27]. To do so, begin by defining the function

$$f(Q;\mathbf{s},\mathbf{a}) = \mathbb{E}_{\mathbf{s}'}\Big[r(\mathbf{s},\mathbf{a},\mathbf{s}') + \gamma \max_{\mathbf{a}'} Q(\mathbf{s}',\mathbf{a}') - Q(\mathbf{s},\mathbf{a}) \,\Big|\, \mathbf{s},\mathbf{a}\Big], \quad (5)$$

and consider an arbitrary everywhere dense distribution $\mathbb{P}(d\mathbf{s}, d\mathbf{a})$ over pairs $(\mathbf{s},\mathbf{a})$ to define the functional

$$L(Q) = \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a}}\Big[f^2(Q;\mathbf{s},\mathbf{a})\Big]. \quad (6)$$

Comparing (5) with (3) permits concluding that $\mathbf{Q}^*$ is the unique function that makes $f(Q;\mathbf{s},\mathbf{a}) = 0$ for all $(\mathbf{s},\mathbf{a})$. It then follows that $\mathbf{Q}^*$ is the only function that makes the functional in (6) take the value $L(Q) = 0$. Since this functional is also nonnegative, we can write the optimal Q function as

$$Q^* = \operatorname*{argmin}_{Q\in\mathcal{B}(\mathcal{S}\times\mathcal{A})} L(Q) \,. \quad (7)$$

Computation of the optimal policy is thus equivalent to solving the optimization problem in (7). This requires a difficult search over all bounded continuous functions $\mathcal{B}(\mathcal{S}\times\mathcal{A})$. We reduce this difficulty 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

$$\text{(i) } \langle f,\kappa((\mathbf{s},\mathbf{a}),\cdot)\rangle_{\mathcal{H}}=f(\mathbf{s},\mathbf{a}),\quad \text{(ii) } \mathcal{H}=\overline{\text{span}\{\kappa((\mathbf{s},\mathbf{a}),\cdot)\}} \tag{8}$$

In (8), property (i) is called the reproducing property. Replacing f by $\kappa((\mathbf{s}',\mathbf{a}'),\cdot)$ in (8) (i) yields the expression $\langle\kappa((\mathbf{s}',\mathbf{a}'),\cdot),\kappa((\mathbf{s},\mathbf{a}),\cdot)\rangle_{\mathcal{H}}=\kappa((\mathbf{s}',\mathbf{a}'),(\mathbf{s},\mathbf{a}))$, the origin of the term "reproducing kernel." Moreover, property (8) (ii) states that functions $f\in\mathcal{H}$ admit a basis expansion in terms of kernel evaluations (9). Function spaces of this type are referred to as reproducing kernel Hilbert spaces (RKHSs).

We may apply the Representer Theorem to transform the functional problem into a parametric one [40], [41]. In the Reproducing Kernel Hilbert Space (RKHS), the optimal Q function takes the following form

$$Q(\mathbf{s},\mathbf{a})=\sum_{n=1}^{N} w_n\kappa((\mathbf{s}_n,\mathbf{a}_n),(\mathbf{s},\mathbf{a})) \tag{9}$$

where $(\mathbf{s}_n,\mathbf{a}_n)$ is a sample of state-action pairs $(\mathbf{s},\mathbf{a})$. $Q\in\mathcal{H}$ is an expansion of kernel evaluations only at observed samples.

One complication of the restriction $\mathcal{B}(\mathcal{S}\times\mathcal{A})$ to the RKHS $\mathcal{H}$ is that this setting requires the cost to be differentiable with Lipschitz gradients, but the definition of $L(Q)$ [cf. (6)] defined by Bellman's equation (4) is non-differentiable due to the presence of the maximization over the Q function. This issue may be addressed by either operating with approximate smoothed gradients of a non-differentiable function [42] or by approximating the non-smooth cost by a smooth one. We adopt the latter approach by replacing the $\max_{\mathbf{a}}Q(\mathbf{s},\mathbf{a}')$ term in (6) by the softmax over continuous range $\mathcal{A}$, i.e.

$$\text{softmax}_{\mathbf{a}'\in\mathcal{A}}Q(\mathbf{s}',\mathbf{a}')=\frac{1}{\eta}\log\int_{\mathbf{a}'\in\mathcal{A}}e^{\eta Q(\mathbf{s}',\mathbf{a}')}d\mathbf{a}' \tag{10}$$

and define the η-smoothed cost $L(Q)$ as the one where the softmax (10) in lieu of the hard maximum in (6). Subsequently, we restrict focus to smoothed cost $L(Q)$.

In this work, we restrict the kernel used to be in the family of universal kernels, such as a Gaussian Gaussian Radial Basis Function(RBF) kernel with constant diagonal covariance Σ,

$$\kappa((\mathbf{s},\mathbf{a}),(\mathbf{s}',\mathbf{a}'))=\exp\{-\frac{1}{2}((\mathbf{s},\mathbf{a})-(\mathbf{s}',\mathbf{a}'))\Sigma((\mathbf{s},\mathbf{a})-(\mathbf{s}',\mathbf{a}')^T\} \tag{11}$$

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 [43].

To apply the Representer Theorem, we require the cost to be coercive in Q [41], which may be satisfied through use of a Hilbert-norm regularizer, so we define the regularized cost functional $J(Q)=L(Q)+(\lambda/2)\|Q\|^2_{\mathcal{H}}$ and solve the regularized problem (7), i.e.

$$Q^*=\underset{Q\in\mathcal{H}}{\text{argmin}}J(Q)=\underset{Q\in\mathcal{H}}{\text{argmin}}L(Q)+\frac{\lambda}{2}\|Q\|^2_{\mathcal{H}}. \tag{12}$$

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 (12). We point out that this is a step back from the original intent of solving (7) to then find optimal policies π^* using (2). This is the case because the assumption we have made about Q^* being representable in the RKHS $\mathcal{H}$ need not be true. More importantly, the functional $J(Q)$ is not convex in Q and there is no guarantee that a local minimum of $J(Q)$ will be the optimal policy Q^*. This is a significant difference relative to policy evaluation problems [27].

III. Stochastic Quasi-Gradient Method

To solve (12), we propose applying a functional variant of stochastic quasi-gradient (SQG) descent to the loss function $J(Q)$ [cf. (12)]. The reasoning for this approach rather than a stochastic gradient method is the nested expectations cause the functional stochastic gradient to be still dependent on a second expectation which is not computable, and SQG circumvents this issue. Then, we apply the Representer Theorem (9) ("kernel trick") to obtain a parameterization of this optimization sequence, which has per-iteration complexity. We then mitigate this untenable complexity growth while preserving optimality using greedy compressive methods, inspired by [34], [27].

To find a stationary point of (12) we use quasi-gradients $\nabla_Q J(Q)$ of the functional $J(Q)$ relative to the function Q in an iterative process. To do so, introduce an iteration index t and let Q_t be the estimate of the stationary point at iteration t. Further consider a random state action pair $(\mathbf{s}_t,\mathbf{a}_t)$ independently and randomly chosen from the distribution $\mathbb{P}(d\mathbf{s},d\mathbf{a})$. Action $\mathbf{a}_t$ is executed from state $\mathbf{s}_t$ resulting in the system moving to state $\mathbf{s}'_t$. This outcome is recorded along with reward $r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}'_t)$ and the action a'_t that maximizes the action-value function Q_t when the system is in state $\mathbf{s}'_t$, i.e.,

$$\mathbf{a}'_t:=\underset{a'}{\text{argmax}}\,Q_t(\mathbf{s}'_t,\mathbf{a}'). \tag{13}$$

The state (S) $\mathbf{s}_t$, action (A) $\mathbf{a}_t$, reward (R) $r(\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}'_t)$, state (S) $\mathbf{s}'_t$, action (A) $\mathbf{a}'_t$ are collectively referred to as the SARSA tuple at time t.

Further consider the expressions for $J(Q)$ in (12) and $L(Q)$ in (6) and exchange order of the expectation and differentiation operators to write the gradient of $J(Q)$ as

$$\nabla_Q J(Q_t)=\mathbb{E}_{\mathbf{s}_t,\mathbf{a}_t}\Big[f(Q_t;\mathbf{s}_t,\mathbf{a}_t)\times\nabla_Q f(Q_t;\mathbf{s}_t,\mathbf{a}_t)\Big]+\lambda Q_t. \tag{14}$$

To compute the directional derivative $\nabla_Q g(Q)$ in (14), we need to address differentiation of the softmax and its approximation properties with respect to the exact maximum, which is done in the following remark.

Remark 1 *(Softmax Gradient Error)* The functional derivative of (10) takes the form

$$\nabla_Q\text{softmax}_{\mathbf{a}'\in\mathcal{A}}Q(\mathbf{s}',\mathbf{a}')=\frac{\int_{\mathbf{a}'\in\mathcal{A}}e^{\eta Q(\mathbf{s}',\mathbf{a}')}\kappa(\mathbf{s}',\mathbf{a}',\cdot)d\mathbf{a}}{\int_{\mathbf{a}'\in\mathcal{A}}e^{\eta Q(\mathbf{s}',\mathbf{a}')}d\mathbf{a}'} \tag{15}$$

by applying Leibniz Rule, Chain Rule, and the reproducing property of the kernel. Moreover, a factor of η cancels. Observe that as $\eta \to \infty$, the softmax becomes closer to the exact (hard) maximum, and the integrals in (15) approach unit, and the only term that remains is $\kappa(\mathbf{s}', \mathbf{a}', \cdot)$. This term may be used in place of (15) to avoid computing the integral, and yields the functional gradient of the exact maximum instead of the softmax. Doing so, however, comes at the cost of computing of the maximizer of the Q function $\mathbf{a}'$.

Observe that to obtain samples of $\nabla_Q J(Q, \mathbf{s}, \mathbf{a}, \mathbf{s}')$ we require two different queries to a simulation oracle: one to approximate the inner expectation over the Markov transition dynamics defined by $\mathbf{s}'$, and one for *each initial pair* $\mathbf{s}, \mathbf{a}$ which defines the outer expectation. This complication, called the "double sampling problem," was first identified in [26], [44], has been ameliorated through use of two time-scale stochastic approximation, which may be viewed as a stochastic variant of quasi-gradient methods [37].

Following this line of reasoning, we build up the total expectation of one of the terms in (14) while doing stochastic descent with respect to the other. In principle, it is possible to build up the expectation of either term in (14), 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 $(\mathbf{s}', \mathbf{a}')$:

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

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 (14) 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 [36], we may extract a tractable parameterization of the infinite dimensional function sequence (18), exploiting properties of the RKHS (8).

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 $\mathcal{O}(t)$ due to the RKHS parametrization, 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)]:

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

Algorithm 1 KQ-Learning

Input: $C, \{\alpha_t, \beta_t\}_{t=0,1,2...}$

1: $Q_0(\cdot) = 0, D_0 = [], w_0 = [], z_0 = 0$
2: **for** $t = 0, 1, 2, \ldots$ **do**
3: Obtain sample $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)$ via exploratory policy
4: Compute maximizing action $\mathbf{a}' = \text{argmax}_{\mathbf{a}} Q_t(\mathbf{s}'_t, \mathbf{a})$
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-grad. step $\tilde{Q}_{t+1} = (1-\alpha_t\lambda) Q_t - \alpha_t z_{t+1} (\gamma \kappa(\mathbf{s}'_t, \mathbf{a}'_t, \cdot) - \kappa(\mathbf{s}_t, \mathbf{a}_t, \cdot))$.
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: Compress function using KOMP with budget $\varepsilon_t = C\alpha_t^2$ $(Q_{t+1}, D_{t+1}, w_{t+1}) = \textbf{KOMP}(\tilde{Q}_{t+1}, \tilde{D}_{t+1}, \tilde{w}_{t+1}, \varepsilon_t)$
9: **end for**
10: **return** Q

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 tune ε_t to ensure both stochastic descent and finite model order – see the next section.

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 a null function $Q_0 = 0$, i.e., 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). KOMP then returns a lower complexity estimate Q_t of $\tilde{Q}_t$ that is ε_t away in $\mathcal{H}$.

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 (13) to define $\mathbf{a}'_t = \text{argmax}_{\mathbf{a}} Q_t(\mathbf{s}'_t, \mathbf{a})$, 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

$$\begin{aligned} &\hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) = \\ &\quad (\gamma\kappa((\mathbf{s}'_t, \mathbf{a}'_t), \cdot) - \kappa((\mathbf{s}_t, \mathbf{a}_t), \cdot)) z_{t+1} + \lambda Q_t \end{aligned} \tag{27}$$

and its sparse-subspace projected variant as

$$\begin{aligned} &\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) = \\ &(Q_t - \mathcal{P}_{\mathcal{H}_{D_{t+1}}}[Q_t - \alpha_t \hat{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)])/\alpha_t \end{aligned} \tag{28}$$

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.,

$$Q_{t+1} = Q_t - \alpha_t \tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) \tag{29}$$

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 : $j^* = \text{argmin}_{j \in \mathcal{I}} \gamma_j$
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 $w = \text{argmin}_{w \in \mathbb{R}^M} \|\tilde{Q}(\cdot) - w^T \kappa_D(\cdot)\|_{\mathcal{H}}$
12: **end if**
13: **end while**
14: **return** V, D, w of model order $M \leq \tilde{M}$ such that $\|Q - \tilde{Q}\|_{\mathcal{H}} \leq \varepsilon_t$

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*

$$\sup_{\mathbf{s}\in\mathcal{S},\mathbf{a}\in\mathcal{A}} \sqrt{\kappa((\mathbf{s},\mathbf{a}),(\mathbf{s},\mathbf{a}))} = K < \infty \tag{30}$$

Moreover, the subspaces $\mathcal{H}_{\mathbf{D}_t}$ are intersected with some finite Hilbert norm ball for each t.

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

$$\mathbb{E}[\delta \,|\, \mathbf{s},\mathbf{a}] = \bar{\delta}\ ,\ \mathbb{E}[(\delta-\bar{\delta})^2] \le \sigma_\delta^2\ ,\ \mathbb{E}[z^2 \,|\, \mathbf{s},\mathbf{a}] \le G_\delta^2 \tag{31}$$

where σ_δ and G_δ are positive scalars, and $\bar{\delta} = \mathbb{E}\{\delta \mid \mathbf{s},\mathbf{a}\}$ is 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:*

$$\mathbb{E}[(\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot) - \kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot))\delta] = \nabla_Q J(Q) \tag{32}$$

$$\mathbb{E}\{\|\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot) - \kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot)\|_{\mathcal{H}}^2 | \mathcal{F}_t\} \le G_Q^2 \tag{33}$$

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

$$\mathbb{E}\{\|\tilde{\nabla}_Q J(Q_t, z_{t+1}; \mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')\|_{\mathcal{H}}^2 | \mathcal{F}_t\} \le \sigma_Q^2 \tag{34}$$

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

$$\|\delta - \bar{\delta}\| \le L_Q \|Q - \tilde{Q}\|_{\mathcal{H}} \tag{35}$$

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. The term related to the maximum of the Q function in the temporal action difference is Lipschitz in the infinity norm since Q is automatically Lipschitz since it belongs to the RKHS. Thus, this term can be related to the Hilbert norm through a constant factor. Hence, (35) is only limits how non-smooth the reward function may be. These are natural extensions of the conditions needed for vector-valued stochastic compositional gradient methods.

Due to Assumption 1 and the use of set projections in (23), we have that Q_t is always bounded in Hilbert norm, i.e., there exists some $0 < D < \infty$ such that

$$\|Q_t\|_{\mathcal{H}} \le D \text{ for all } t\ . \tag{36}$$

With these technical conditions, we can derive a coupled stochastic descent-type relationship of Algorithm 1 and then apply the Coupled Supermartingale Theorem [45][Lemma 6] to establish convergence, which we state next.

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:*

$$\sum_{t=1}^{\infty} \alpha_t = \infty, \sum_{t=1}^{\infty} \beta_t = \infty, \sum_{t=1}^{\infty} \alpha_t^2 + \beta_t^2 + \frac{\alpha_t^2}{\beta_t} < \infty, \varepsilon_t = C\alpha_t^2 \tag{37}$$

Then $\|\nabla_Q J(Q)\|_{\mathcal{H}}$ converges to null with probability 1, and hence Q_t attains a stationary point of (12)*. In particular, the limit of Q_t achieves the regularized Bellman fixed point restricted to the RKHS.*

See Appendix B.

Theorem 1 establishes that Algorithm 1 converges almost surely to a stationary solution of the problem (12) 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 particular, by modifying the above exact convergence result for diminishing learning rates to one in which they are kept constant, we are able to keep constant compression budgets as well, and establish convergence to a neighborhood as well as the finiteness of the model order of Q, as we state next.

Theorem 2 *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-sizes are chosen as constant such that $0 < \alpha < \beta < 1$, with $\varepsilon = C\alpha^2$ and the parsimony constant $C > 0$ is positive. Then the Bellman error converges to a neighborhood in expectation, i.e.:*

$$\liminf_{t\to\infty} \mathbb{E}[J(Q_t)] \le \mathcal{O}\left(\frac{\alpha\beta}{\beta-\alpha}\left[1 + \sqrt{1 + \frac{\beta-\alpha}{\alpha\beta}\left(\frac{1}{\beta} + \frac{\beta^2}{\alpha^2}\right)}\right]\right) \tag{38}$$

See Appendix C.

The expression on the right-hand side of (38) is a complicated posynomial of α and β, but is positive provided $\beta > \alpha$, and for a fixed β increases as α increases. This means that more aggressive selections of α, for a given β, yield a larger limiting lower bound on the Bellman error. A simple example

which satisfies the constant step-size conditions $0 < \alpha < \beta < 1$ is $\beta = \alpha + \iota$ for some small constant $\iota > 0$. This is consistent with the diminishing step-size conditions where $\alpha_t/\beta_t \to 0$ means that α_t must be smaller than β_t which is in $(0,1)$.

An additional salient feature of the parameter choice given in Theorem 2 is that [27][Corollary 1] applies, and thus we may conclude that the Q-function parameterization is at-worst finite during learning when used with constant step-sizes and compression budget. In subsequent sections, we investigate the empirical validity of the proposed approach on two autonomous control tasks: the Inverted Pendulum and Continuous Mountain Car, for which observe consistent convergence in practice. To do so, first some implementation details of Algorithm 1 must be addressed.

V. Practical Considerations

The convergence guarantees for Algorithm 1 require sequentially observing state-action-next-state triples $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}'_t)$ independently and identically distributed. Doing so, however, only yields convergence toward a stationary point, which may or may not be the optimal Q function. To improve the quality of stationary points to which we converge, it is vital to observe states that yield reward (an instantiation of the explore-exploit tradeoff). To do so, we adopt a standard practice in reinforcement learning which is to bias actions towards states that may accumulate more reward.

The method in which we propose to bias actions is by selecting them according to the current estimate of the optimal policy, i.e., the greedy policy. However, when doing so, the KQ-Learning updates (18) computed using greedy samples $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}'_t)$ are composed of two points nearby in $\mathcal{S} \times \mathcal{A}$ space. These points are then evaluated by kernels and given approximately equal in opposite weight. Thus, this update is immediately pruned away during the execution of KOMP [32], [46], [34]. In order to make use of the greedy samples and speed up convergence, we project the functional update onto just one kernel dictionary element, resulting in the update step:

$$\tilde{Q}_{t+1} = (1-\alpha_t\lambda)Q_t(\cdot) + \alpha_t(\kappa(\mathbf{s}_t, \mathbf{a}_t, \cdot))z_{t+1}) \tag{39}$$

The resulting procedure is summarized as Algorithm 3. First, trajectory samples are obtained using a greedy policy. Then, the temporal-action difference is computed and averaged recursively. Finally, we update the Q function via (39) and compress it using Algorithm 2.

ρ-Greedy Actions and Hybrid Update To address the explore-exploit trade-off, we use an ρ-greedy policy [47]: with probability ρ we select a random action, and select a greedy action with probability $1-\rho$. We adopt this approach with ρ decreasing linearly during training, meaning that as time passes more greedy actions are taken.

The algorithm when run with a ρ-greedy policy is described as the *Hybrid* algorithm, which uses Algorithm 1 when exploratory actions are taken and Algorithm 3 for greedy actions. Practically, we find it useful to judiciously use training examples, which may be done with a data buffer. Thus, the hybrid algorithm is as follows: First, we accumulate trajectory samples in a buffer. Along with the $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}'_t)$ sample, we store an indicator whether $\mathbf{a}_t$ was an exploratory action or greedy with respect to Q_{t-1}. Then, samples are drawn at random from the buffer for training. We explore two different methods for obtaining samples from the buffer: uniformly at random, and prioritized sampling, which weighs each sample in the buffer by its observed Bellman error. For greedy actions, we use the update in (39), and for exploratory actions, we use the KQ-learning update from 1. Finally, we use KOMP to compress the representation of the Q function.

Algorithm 3 Semi-Gradient Greedy KQ-Learning

Input: $C, \{\alpha_t, \beta_t\}_{t=0,1,2\ldots}$
$Q_0(\cdot) = 0, D_0 = [], w_0 = [], z_0 = 0$
for $t = 0, 1, 2, \ldots$ **do**
  Obtain sample $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t)$ via greedy policy
  Compute maximizing action:
  $\mathbf{a}'_t = \pi_t(\mathbf{s}'_t) = \text{argmax}_{\mathbf{a}} Q_t(\mathbf{s}'_t, \mathbf{a})$
  Update temporal action diff. δ_t and aux. seq. z_{t+1}
  $\delta_t = r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}'_t) + \gamma Q_t(\mathbf{s}'_t, \mathbf{a}'_t) - Q_t(\mathbf{s}_t, \mathbf{a}_t)$
  $z_{t+1} = (1-\beta_t)z_t + \beta_t\delta_t$.
  Compute update step
  $\tilde{Q}_{t+1} = (1-\alpha_t\lambda)Q_t(\cdot) + \alpha_t z_{t+1}\kappa(\mathbf{s}_t, \mathbf{a}_t, \cdot)$.
  Update dictionary $\tilde{D}_{t+1} = [D_t, (\mathbf{s}, \mathbf{a})]$,
  weights $\tilde{w}_{t+1} = [(1-\alpha_t\lambda)w_t, \alpha_t z_{t+1}]$.
  Compress function using KOMP with budget $\varepsilon_t = C\alpha_t^2$
  $(Q_{t+1}, D_{t+1}, w_{t+1}) = \textbf{KOMP}(\tilde{Q}_{t+1}, \tilde{D}_{t+1}, \tilde{w}_{t+1}, \varepsilon_t)$
end for
return Q

Maximizing the Q Function In order to implement Algorithm 3, we apply simulated annealing [48] to evaluate the instantaneous maximizing action $\mathbf{a}_t = \text{argmax}_{\mathbf{a}} Q_t((\mathbf{s}_t, \mathbf{a})$. For a general reproducing kernel $\kappa(\cdot,\cdot)$, maximizing over a weighted sum of kernels is a non-convex optimization problem, so we get stuck in undesirable stationary points [49]. To reduce the chance that this undesirable outcome transpires, we use simulated annealing. First, we sample actions $\mathbf{a}$ uniformly at random from the action space. Next, we use gradient ascent to refine our estimate of the global maximum of Q for state $\mathbf{s}$. We use the Gaussian Radial Basis Function(RBF) kernel (11), so the Q function is differentiable with respect to an arbitrary action $\mathbf{a}$:

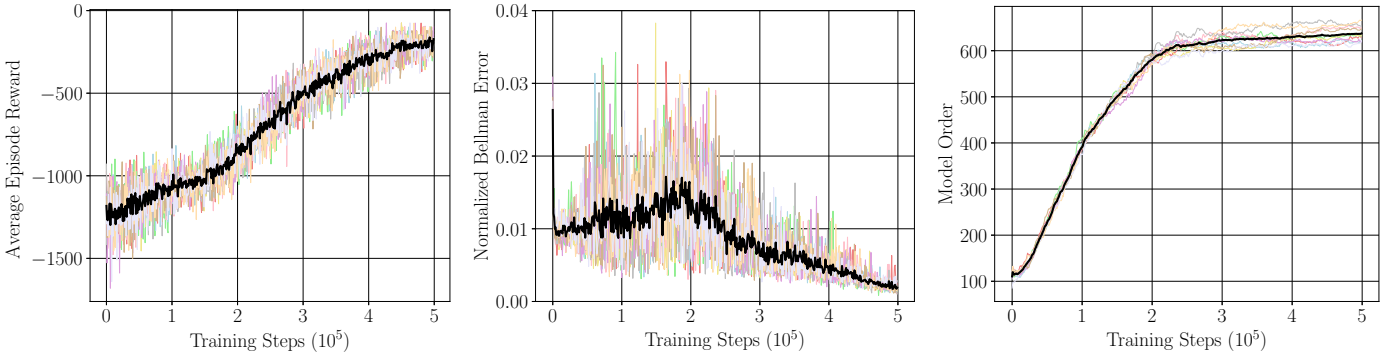
$$(\nabla_{\mathbf{a}} Q)(\mathbf{s}, \mathbf{a}) = Q(\mathbf{s}, \mathbf{a}) \sum_{m=1}^{M} w_m \Sigma_{\mathbf{a}} (\mathbf{a} - \mathbf{a}_m)^T \tag{40}$$

and that gradient evaluations are cheap: typically their complexity scales with the model order of the Q function which is a kept under control using Algorithm 2.

Remark 2 Observe that (39) bears a phantom resemblance to Watkins' Q-Learning algorithm [9]; however, it is unclear how to extend [9] to continuous MDPs where function approximation is required. In practice, using (39) for all updates, we observe globally steady policy learning and convergence of Bellman error, suggesting a link between (39) and stochastic fixed point methods [13], [14]. This link is left to future investigation. For now, we simply note that stochastic fixed point iteration is fundamentally different than stochastic descent

	Environment	Algorithm	Replay Buffer	Policy	Steps	α	β	C	Kernel Σ	Order	Loss	Rewards
1	Inv. Pendulum	KQ	Yes	Exploratory	100K	0.25	1.00	2.00	[0.5,0.5,2,0.5]	137.17	0.79	-1194.35
2	Inv. Pendulum	KQ	Yes	ρ-greedy	100K	0.25	1.00	2.00	[0.5,0.5,2,0.5]	111.71	22.26	-1493.65
3	Inv. Pendulum	Hybrid	Yes	ρ-greedy	500K	0.25	1.00	2.00	[0.5,0.5,2,0.5]	**636.2**	**0.99**	**-160.01**
4	Inv. Pendulum	SG	Yes	ρ-greedy	200K	0.25	1.00	2.00	[0.5,0.5,2,0.5]	**749.75**	**2.92**	**-150.36**
5	Inv. Pendulum	KQ	No	Exploratory	100K	0.25	1.00	2.00	[0.5,0.5,2,0.5]	134.14	0.72	-1258.43
6	Inv. Pendulum	KQ	No	ρ-greedy	100K	0.25	1.00	2.00	[0.5,0.5,2,0.5]	257.71	14.5	-1258.45
7	Inv. Pendulum	Hybrid	No	ρ-greedy	500K	0.25	1.00	2.00	[0.5,0.5,2,0.5]	684.39	0.61	-180.37
8	Inv. Pendulum	SG	No	ρ-greedy	200K	0.25	1.00	2.00	[0.5,0.5,2,0.5]	772.69	1.88	-247.17
9	Cont. M. Car	KQ	Prioritized	Exploratory	100K	0.25	1.00	0.10	[0.8,0.07,1.0]	44.54	0.41	-20.61
10	Cont. M. Car	KQ	Prioritized	ρ-greedy	500K	0.25	1.00	0.10	[0.8,0.07,1.0]	67.0	0.92	85.43
11	Cont. M. Car	Hybrid	Prioritized	ρ-greedy	500K	0.25	1.00	0.10	[0.8,0.07,1.0]	71.22	0.76	94.72
12	Cont. M. Car	SG	Prioritized	ρ-greedy	500K	0.25	1.00	0.10	[0.8,0.07,1.0]	87.56	0.81	94.75
13	Cont. M. Car	KQ	No	Exploratory	100K	0.25	1.00	0.10	[0.8,0.07,1.0]	36.53	0.29	-21.41
14	Cont. M. Car	KQ	No	ρ-greedy	500K	0.25	1.00	0.10	[0.8,0.07,1.0]	58.42	0.21	80.92
15	Cont. M. Car	Hybrid	No	ρ-greedy	500K	0.25	1.00	0.10	[0.8,0.07,1.0]	**57.26**	**0.48**	**94.96**
16	Cont. M. Car	SG	No	ρ-greedy	500K	0.25	1.00	0.10	[0.8,0.07,1.0]	**63.13**	**0.42**	**94.83**

TABLE I: A summary of parameter selection details for our comparison of KQ-Learning(KQ), Hybrid, and Semi-Gradient(SG) methods. In right-most column, we display the limiting model order, training loss (Bellman error) and accumulation of rewards during training. The best results for each problem setting are bolded for emphasis, which "solve" the problem according to reward benchmarks set by OpenAI. We observe the replay buffer improves learning in the Pendulum domain but yields little benefit in the Mountain Car problem. Interestingly, the Hybrid algorithm in the Pendulum domain attains a smaller training Bellman error but less rewards than the SG approach.



(a) Average Training Reward (b) Training 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 for the Pendulum domain using the Hybrid algorithm with a replay buffer, Row 3 in Table I. Fig. 1a shows the average reward obtained by the ρ-greedy policy during training. Fig. 1b shows the Bellman error for training samples (6) 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 700 on average. Overall, we solve Pendulum with a model complexity reduction by orders of magnitude relative to existing methods [17], [50], with a much smaller standard deviation around the average reward accumulation, meaning that these results are replicable.

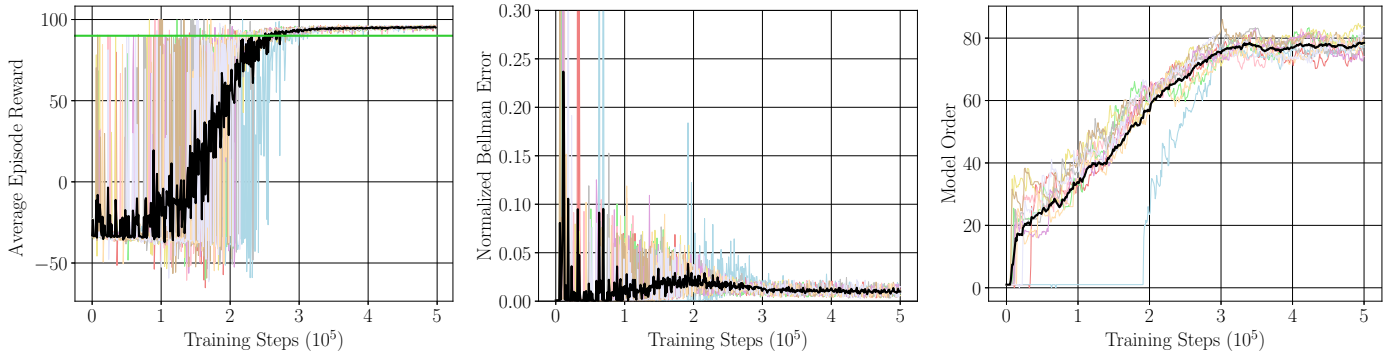
methods which rely on the construction of supermartingales, so results from the previous section do not apply to (39). Moreover, this update has also been referred to as a temporal difference "semi-gradient" update in Chapters 9-10 of [5].

VI. Experiments

We shift focus to experimentation of the methods developed and analyzed in the previous sections. Specifically, we benchmark the proposed algorithms on two classic control problems, the Inverted Pendulum [39], [41] and the Continuous Mountain Car [38], which are featured in OpenAI Gym [51].

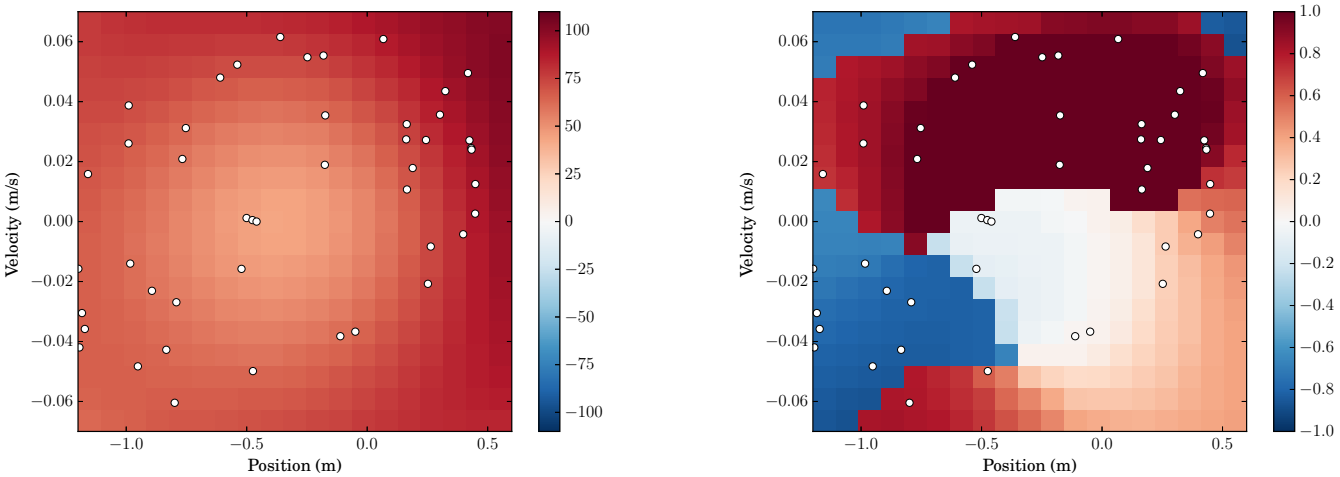
In the Inverted Pendulum problem, the state space is $p = 3$ dimensional, consisting of the sine of the angle of the pendulum, the cosine of the angle, and the angular velocity, bounded within $[-1.0, 1.0]$, $[-1.0, 1.0]$, and $[-8.0, 8.0]$ respectively. The action space is $q = 1$ dimensional: joint effort, within the interval $[-2.0, 2.0]$. The reward function is $r(\theta, \dot{\theta}, a) = -(\theta^2 + 0.1\dot{\theta}^2 + 0.001a^2)$, where θ is the angle of the pendulum relative to vertical, and $\dot{\theta}$ is the angular velocity. The goal of the problem is to balance the pendulum at the unstable equilibrium where $\theta = 0$.

In the Continuous Mountain Car 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



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

Fig. 2: Results of 10 experiments over 500,000 training steps were averaged (black curve) to demonstrate the learning progress for Continuous Mountain Car using the Hybrid algorithm with no replay buffer, Row 15 in Table I. Fig. 2a shows the average reward obtained by the ρ-greedy policy during training. An average reward over 90 (green) indicates that we have solved Continuous Mountain Car, steering towards the goal location. Fig. 2b shows the normalized Bellman error during training, which converges to a small non-zero value. Fig. 2c shows the number of points parameterizing the kernel dictionary of Q during training, which remains under 80 on average. Overall, we solve Continuous Mountain Car with a complexity reduction by orders of magnitude relative to existing methods[17], [50]. We observe that learning progress has higher variance, which we hypothesize is related to the sparsity of the reward signal.



(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. 3: For the mountain car problem, the learned Q-function is easily interpretable: we may visualize the value function, $V(s) = \max_a Q(s,a)$ (3a) and corresponding policy $\pi(s) = \text{argmax}_a Q(s,a)$ (3b). In Fig. 3a, 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. 3b, 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.

the interval $[-1,1]$. The reward function is 100 when the car reaches the goal at position 0.45, and $-0.1a^2$ for any action a. For each training episode, the start position of the car was initialized uniformly at random in the range $[-0.6, 0.4]$.

For all experiments with the Inverted Pendulum and the Continuous Mountain Car problems, we used Gaussian kernels with a fixed non-isotropic bandwidth. 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, and are summarized in Table I.

We investigate two methods for exploration as the agent traverses the environment. When using an exploratory policy, actions are selected uniformly at random from the action space. When using an ρ-greedy policy, we select actions randomly with probability 1, which linearly decays to 0.1 after 10^5 exploratory training steps. In addition, we explore the use of a replay buffer. This method re-reveals past data to the agent uniformly at random. For the Mountain Car problem, we also use prioritized memory which replays samples based on the magnitude of their temporal action difference.

A comprehensive summary of our experimental results

may be found in Table I. We bold which methods perform best across many different experimental settings. Interestingly, playback buffers play a role in improving policy learning in the Pendulum domain but not for Mountain Car, suggesting that their merit demands on the reward structure of the MDP.

We spotlight the results of this experiment in the Pendulum domain for the Hybrid algorithm in Figure 1: here we plot the normalized Bellman test error Fig. 1b, defined by the sample average approximation of (6) 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 -200, which is comparable to top entries on the OpenAI Leaderboard [51], such as Deep Deterministic Policy Gradient [50] . Moreover, we obtain this result with a complexity reduction by orders of magnitude relative to existing methods for Q-function and policy representation. This trend is corroborated for the Continuous Mountain Car in Figure 2: the normalized Bellman error converges and the model complexity remains moderate. Also, observe that the interval average rewards approach 90, which is the benchmark used to designate a policy as "solving" Continuous Mountain Car.

Additionally, few heuristics are required to ensure KQ-Learning converges in contrast to neural network approaches to Q-learning. One shortcoming of our implementation is its sample efficiency, which could improved through a mini-batch approach. Alternatively, variance reduction, acceleration, or Quasi-Newton methods would improve the learning rate.

A feature of our method is the interpretability of the resulting Q function, which we use to plot the value function (3a) and policy (3b). One key metric is the coverage of the kernel points in the state-action space. We can make conclusions about the importance of certain parts of the space for obtaining as much value as possible by the density of the model points throughout the space. This may have particular importance in mechanical or econometric applications, where the model points represent physical phenomena or specific events in financial markets.

VII. Conclusion

In this paper, we extended the nonparametric optimization approaches in [27] from policy evaluation to policy learning in continuous Markov Decision Problems. In particular, we reformulated the task of policy learning defined by the Bellman optimality equation as a non-convex function-valued stochastic program with nested expectations. We hypothesize that the Bellman fixed point belongs to a reproducing Kernel Hilbert Space, motivated by their efficient semi-parametric form. By applying functional stochastic quasi-gradient method operating in tandem with greedily constructed subspace projections, we derived a new efficient variant of Q learning which is guaranteed to converge almost surely in continuous spaces, one of the first results of this type.

Unlike the policy evaluation setting, in policy learning we are forced to confront fundamental limitations associated with non-convexity and the explore-exploit tradeoff. To do so, we adopt a hybrid policy learning situation in which some actions are chosen greedily and some are chosen randomly. Through careful tuning of the proportion of actions that are greedy versus exploratory, we are able to design a variant of Q learning which learns good policies on some benchmark tasks, namely, the Continuous Mountain Car and the Inverted Pendulum, with orders of magnitude fewer training examples than existing approaches based on deep learning. Further, owing to the kernel parameterization of our learned Q functions, they are directly interpretable: the training points which are most vital for representing the minimal Bellman error action-value function are retained and automatically define its feature representation.

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, 2018.

[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. 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.

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

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

[14] 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.

[15] 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.

[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 IEEE TAC Dec. 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] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," in *Advances in neural information processing systems*, 2002, pp. 785–792.
[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] A. W. Moore, "Efficient memory-based learning for robot control," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-209, Nov. 1990. [Online]. Available: http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-209.pdf
[39] K. Yoshida, "Swing-up control of an inverted pendulum by energy-based methods," in *American Control Conference, 1999. Proceedings of the 1999*, vol. 6. IEEE, 1999, pp. 4045–4047.
[40] 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.
[41] 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.
[42] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005.
[43] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *Journal of Machine Learning Research*, vol. 7, no. Dec, pp. 2651–2667, 2006.
[44] V. S. Borkar, "Stochastic approximation with two time scales," *Systems & Control Letters*, vol. 29, no. 5, pp. 291–294, 1997.
[45] M. Wang and D. P. Bertsekas, "Incremental constraint projection-proximal methods for nonsmooth convex optimization," *SIAM Journal on Optimization (to appear)*, 2014.
[46] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, no. 1, pp. 165–187, 2002.
[47] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvari, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 287–308, 2000.
[48] E. Aarts and J. Korst, "Simulated annealing and boltzmann machines," 1988.
[49] 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.
[50] 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.
[51] Openai gym - continuous mountain car. [Online]. Available: https://gym.openai.com/envs/MountainCarContinuous-v0/
[52] D. P. Bertsekas and S. Shreve, *Stochastic optimal control: the discrete-time case*, 2004.
[53] W. Rudin, *Principles of mathematical analysis*, 3rd ed. New York: McGraw-Hill Book Co., 1976, international Series in Pure and Applied Mathematics.

Alec Koppel began as a Research Scientist at the U.S. Army Research Laboratory in the Computational and Information Sciences Directorate in September of 2017. He completed his Master's degree in Statistics and Doctorate in Electrical and Systems Engineering, both at the University of Pennsylvania (Penn) in August of 2017. He is also a participant in the Science, Mathematics, and Research for Transformation (SMART) Scholarship Program sponsored by the American Society of Engineering Education. Before coming to Penn, he completed his Master's degree in Systems Science and Mathematics and Bachelor's Degree in Mathematics, both at Washington University in St. Louis (WashU), Missouri. His research interests are in the areas of signal processing, optimization and learning theory. His current work focuses on optimization and learning methods for streaming data applications, with an emphasis on problems arising in autonomous systems. He co-authored a paper selected as a Best Paper Finalist at the 2017 IEEE Asilomar Conference on Signals, Systems, and Computers.

Ekaterina Tolstaya is a doctoral student in the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, and a National Science Foundation Graduate Research Fellow. She received B.Sc. degrees in Electrical Engineering and Computer Science from University of Maryland, College Park, MD, in 2016, and the M.S.E. degree in Robotics from University of Pennsylvania, Philadelphia, PA, in 2017. Her research interests include reinforcement learning, aerial robotics, and multi-agent systems.

Ethan A. Stump received the B.S. degree from the Arizona State University, Tempe, and the M.S. and PhD degrees from the University of Pennsylvania, Philadelphia, all in mechanical engineering. He is a researcher within the U.S. Army Research Laboratory's Computational and Information Sciences Directorate, where he has worked on developing mapping and navigation technologies to enable baseline autonomous capabilities for teams of ground robots and on developing controller synthesis for managing the deployment of multi-robot teams to perform repeating tasks such as persistent surveillance by tying them formal task specifications.

Alejandro Ribeiro received the B.Sc. degree in electrical engineering from the Universidad de la Republica Oriental del Uruguay, Montevideo, in 1998 and the M.Sc. and Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, the University of Minnesota, Minneapolis in 2005 and 2007. From 1998 to 2003, he was a member of the technical staff at Bellsouth Montevideo. After his M.Sc. and Ph.D studies, in 2008 he joined the University of Pennsylvania (Penn), Philadelphia, where he is currently the Rosenbluth Associate Professor at the Department of Electrical and Systems Engineering. His research interests are in the applications of statistical signal processing to the study of networks and networked phenomena. His current research focuses on wireless networks, network optimization, learning in networks, networked control, robot teams, and structured representations of networked data structures. Dr. Ribeiro received the 2012 S. Reid Warren, Jr. Award presented by Penn's undergraduate student body for outstanding teaching, the NSF CAREER Award in 2010, and student paper awards at the 2013 American Control Conference (as adviser), as well as the 2005 and 2006 International Conferences on Acoustics, Speech and Signal Processing. Dr. Ribeiro is a Fulbright scholar and a Penn Fellow.

VIII. Appendices

Appendix A: Proof of Auxiliary Results

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*

$$\|\tilde{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}'_t)-\hat{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}'_t)\|_{\mathcal{H}}\leq\frac{\varepsilon_t}{\alpha_t} \tag{41}$$

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 (27) and (28)

$$\begin{aligned}\|\tilde{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}'_t)-\hat{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}'_t)\|_{\mathcal{H}}=\\ \|(Q_t-\mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}[Q_t-\alpha_t\hat{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}'_t)])/\alpha_t\\ -\hat{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}'_t)\|^2_{\mathcal{H}}\end{aligned} \tag{42}$$

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

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

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 (41). ■

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

$$\mathbb{E}[\|Q_{t+1}-Q_t\|^2_{\mathcal{H}}|\mathcal{F}_t]\leq 2\alpha_t^2(G_\delta^2G_Q^2+\lambda D^2)+2\varepsilon_t^2 \tag{44}$$

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*

$$\begin{aligned}\mathbb{E}[(z_{t+1}-\bar{\delta}_t)^2\,|\,\mathcal{F}_t]&\leq(1-\beta_t)(z_t-\bar{\delta}_{t-1})^2\\ &\quad+\frac{L_Q}{\beta_t}\|Q_t-Q_{t-1}\|^2_{\mathcal{H}}+2\beta_t^2\sigma_\delta^2\end{aligned} \tag{45}$$

iii. Algorithm 1 generates a sequence of Q-functions that satisfy the stochastic descent property with respect to the Bellman error $J(Q)$ *[cf.* (12)*]:*

$$\begin{aligned}\mathbb{E}[J(Q_{t+1})\,|\,\mathcal{F}_t]&\leq J(Q_t)-\alpha_t\left(1-\frac{\alpha_tG_Q^2}{\beta_t}\right)\|\nabla_Q J(Q)\|^2\\ &\quad+\frac{\beta_t}{2}\mathbb{E}[(\bar{\delta}_t-z_{t+1})^2\,|\,\mathcal{F}_t]+\frac{L_Q\sigma_Q^2\alpha_t^2}{2}\\ &\quad+\varepsilon_t\|\nabla_Q J(Q_t)\|_{\mathcal{H}}\,,\end{aligned} \tag{46}$$

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}

$$\|Q_{t+1}-Q_t\|^2_{\mathcal{H}}=\alpha_t^2\|\tilde{\nabla}_Q J(Q_t,z_{t+1};(\mathbf{s}_t,\mathbf{a}_t),(\mathbf{s}'_t,\mathbf{a}'_t)\|^2_{\mathcal{H}} \tag{47}$$

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 (47) and apply the triangle inequality $(a+b)^2\leq 2a^2+2b^2$ which holds for any $a,b>0$.

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

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

$$\begin{aligned}&\mathbb{E}[\|Q_{t+1}-Q_t\|^2_{\mathcal{H}}]\,|\,\mathcal{F}]\\ &\quad=2\alpha_t^2\mathbb{E}[\|\hat{\nabla}_Q J(Q_t,z_{t+1};(\mathbf{s}_t,\mathbf{a}_t),(\mathbf{s}'_t,\mathbf{a}'_t)\|^2_{\mathcal{H}}\,|\,\mathcal{F}_t]+2\varepsilon_t^2\end{aligned} \tag{49}$$

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 (49) as

$$\begin{aligned}&\mathbb{E}[\|Q_{t+1}-Q_t\|^2_{\mathcal{H}}\,|\,\mathcal{F}_t]\\ &\quad\leq 2\alpha_t^2\mathbb{E}\{\|\gamma\kappa((\mathbf{s}'_t,\mathbf{a}'_t),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot)\|^2_{\mathcal{H}}\\ &\qquad\times\mathbb{E}[z_{t+1}^2\,|\,\mathbf{s}_t,a_t]\,|\,\mathcal{F}_t\}+2\alpha_t^2\lambda\|Q_t\|^2_{\mathcal{H}}+2\varepsilon_t^2\end{aligned} \tag{50}$$

Now, use the fact that z_{t+1} has a finite second conditional moment [cf. (31)], yielding

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

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 (31) and that the Q function sequence is bounded (36) to write:

$$\mathbb{E}[\|Q_{t+1}-Q_t\|_{\mathcal{H}}^2 \mid \mathcal{F}_t] \leq 2\alpha_t^2(G_\delta^2 G_V^2+\lambda^2 D^2)+2\varepsilon_t^2 \tag{52}$$

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:

$$\begin{aligned}z_{t+1}-\bar{\delta}_t+e_t =&(1-\beta_t)z_t+\beta_t\delta_t-[(1-\beta_t)\bar{\delta}_t+\beta_t\bar{\delta}_t] \\ &+(1-\beta_t)(\bar{\delta}_t-\bar{\delta}_{t-1})\end{aligned} \tag{53}$$

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 (53). 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 (53), using this simplification, yields

$$\begin{aligned}(z_{t+1}-\bar{\delta}_t+e_t)^2 \\ =(1-\beta_t)^2(z_t-\bar{\delta}_{t-1})^2+\beta_t^2(\delta_t-\bar{\delta}_t)^2 \\ +2(1-\beta_t)\beta_t(z_t-\bar{\delta}_{t-1})(\delta_t-\bar{\delta}_t)\end{aligned} \tag{54}$$

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

$$\begin{aligned}\mathbb{E}[(z_{t+1}-\bar{\delta}_t+e_t)^2 \mid \mathcal{F}_t] \\ =(1-\beta_t)^2(z_t-\bar{\delta}_{t-1})^2+\beta_t^2\mathbb{E}[(\delta_t-\bar{\delta}_t)^2 \mid \mathcal{F}_t] \\ +2(1-\beta_t)\beta_t(z_t-\bar{\delta}_{t-1})\mathbb{E}[(\delta_t-\bar{\delta}_t) \mid \mathcal{F}_t]\end{aligned} \tag{55}$$

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

$$\mathbb{E}[(z_{t+1}-\bar{\delta}_t+e_t)\big|\mathcal{F}_t]=(1-\beta_t)^2(z_t-\bar{\delta}_{t-1})^2+\beta_t^2\sigma_\delta^2 \tag{56}$$

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

$$(z_{t+1}-\bar{\delta}_t)^2\leq(1+\beta_t)(z_{t+1}-\bar{\delta}_t+e_t)^2+\left(1+\frac{1}{\beta_t}\right)e_t^2 \tag{57}$$

Observe that (57) 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 (57) conditional on $\mathcal{F}_t$ and substitute (56) 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 yields

$$\begin{aligned}\mathbb{E}[(z_{t+1}-\bar{\delta}_t)^2 \mid \mathcal{F}_t] \\ =(1+\beta_t)[(1-\beta_t)^2(z_t-\bar{\delta}_{t-1})^2+\beta_t^2\sigma_\delta^2]+\left(\frac{1+\beta_t}{\beta_t}\right)e_t^2\end{aligned} \tag{58}$$

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

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

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

$$(1-\beta_t)|(\bar{\delta}_t-\bar{\delta}_{t-1})|\leq(1-\beta_t)L_Q\|Q_t-Q_{t-1}\|_{\mathcal{H}}\,, \tag{60}$$

where we apply the Lipschitz continuity of the temporal difference with respect to the action-value function [cf. (35)]. Substitute the right-hand side of (60) and simplify the expression in the last term as $(1-\beta_t^2)/\beta_t\leq 1/\beta_t$ to conclude (46). ■

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:

$$\begin{aligned}J(Q_{t+1})\leq J(Q_t)+\langle\nabla J(Q_t),Q_{t+1}-Q_t\rangle_{\mathcal{H}} \\ +\frac{L_Q}{2}\|Q_{t+1}-Q_t\|_{\mathcal{H}}^2\,.\end{aligned} \tag{61}$$

Substitute the fact that the difference between consecutive action-value functions is the projected quasi-stochastic gradient $Q_{t+1}-Q_t=-\alpha_t\tilde{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')$ (29) into (61) .

$$\begin{aligned}J(Q_{t+1})\leq&J(Q_t)-\alpha_t\langle\nabla J(Q_t),\tilde{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')\rangle_{\mathcal{H}} \\ &+\frac{L_Q\alpha_t^2}{2}\|\tilde{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')\|_{\mathcal{H}}^2\,.\end{aligned} \tag{62}$$

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, (27) and (28), respectively. Now add and subtract the inner-product of the functional gradient of J with the stochastic gradient, scaled by the step-size $\alpha_t\langle\nabla J(Q_t),\hat{\nabla}_Q J(Q_t,z_{t+1};\mathbf{s}_t,\mathbf{a}_t,\mathbf{s}_t')\rangle_{\mathcal{H}}$, and $\alpha_t\|\nabla_Q J(Q_t)\|^2$ into above expression and gather terms.

$$\begin{aligned}J(Q_{t+1})\leq J(Q_t)-\alpha_t\|\nabla_Q J(Q)\|^2+\frac{L_Q\alpha_t^2}{2}\|\tilde{\nabla}_Q J(Q_t)\|_{\mathcal{H}}^2 \\ -\alpha_t\langle\nabla J(Q_t),\tilde{\nabla}_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\rangle_{\mathcal{H}} \\ +\alpha_t\langle\nabla J(Q_t),\nabla J(Q_t)-\hat{\nabla}_Q J(Q_t)\rangle_{\mathcal{H}}\end{aligned} \tag{63}$$

Observe that the last two terms on the right-hand side of (63) 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 the Law of Total Expectation and Assumptions 2 - 3.

We proceed to address the second term on the right-hand side of (63) by applying Cauchy-Schwarz to write

$$\begin{aligned}&|-\alpha_t\langle\nabla J(Q_t),\tilde{\nabla}_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\rangle_{\mathcal{H}}|\\&\quad\leq\alpha_t\|\nabla J(Q_t)\|_{\mathcal{H}}\|\tilde{\nabla}_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\|_{\mathcal{H}}\end{aligned}\tag{64}$$

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 (64) and cancel the factor of α_t:

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

Next, we address the last term on the right-hand side of (63). 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 (32):

$$\begin{aligned}&\mathbb{E}\big[\nabla J(Q_t),\nabla_Q J(Q_t)-\hat{\nabla}_Q J(Q_t)\rangle_{\mathcal{H}}\,\big|\,\mathcal{F}_t\big]\\&=\Big\langle\nabla_Q J(Q_t),\mathbb{E}[(\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot))(\bar{\delta}_t-z_{t+1})\big|\mathcal{F}_t]\Big\rangle_{\mathcal{H}}\end{aligned}\tag{66}$$

In (66), 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 (66), by using (32), 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

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

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:

$$\begin{aligned}&\|\nabla_Q J(Q_t)\|_{\mathcal{H}}\mathbb{E}[\|(\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot))\|_{\mathcal{H}}|\bar{\delta}_t-z_{t+1}\big|\mathcal{F}_t]\Big\rangle_{\mathcal{H}}\\&\leq\frac{\beta_t}{2}\mathbb{E}[(\bar{\delta}_t-z_{t+1})^2\mid\mathcal{F}_t]\\&+\frac{\alpha_t^2}{2\beta_t}\|\nabla J(Q_t)\|_{\mathcal{H}}^2\mathbb{E}[\|\gamma\kappa((\mathbf{s}_t',\mathbf{a}_t'),\cdot)-\kappa((\mathbf{s}_t,\mathbf{a}_t),\cdot)\|_{\mathcal{H}}^2\,\big|\,\mathcal{F}_t]\end{aligned}\tag{68}$$

To (68), we apply Assumption 3 regarding the finite second conditional of the difference of reproducing kernel maps (33) to the second term, which when substituted into the right-hand side of the expectation of (63) conditional on $\mathcal{F}_t$, yields

$$\begin{aligned}\mathbb{E}[J(Q_{t+1})\,|\,\mathcal{F}_t]\leq J(Q_t)-\alpha_t\left(1-\frac{\alpha_t G_Q^2}{\beta_t}\right)\|\nabla_Q J(Q)\|^2\\+\frac{\beta_t}{2}\mathbb{E}[(\bar{\delta}_t-z_{t+1})^2\mid\mathcal{F}_t]+\frac{L_Q\sigma_Q^2\alpha_t^2}{2}\\+\varepsilon_t\|\nabla_Q J(Q_t)\|_{\mathcal{H}}\,,\end{aligned}\tag{69}$$

where we have also applied the fact that the projected stochastic quasi-gradient has finite conditional variance (34) and gathered like terms to conclude (46). ■

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.

Appendix B: Proof of Theorem 1

We use the relations established in Lemma 1 to construct a coupled supermatingale of the form 2. First, we state the following lemma regarding coupled sequences of conditionally decreasing stochastic processes called the coupled supermartingale lemma, stated as:

Lemma 2 *(Coupled Supermartingale Theorem) [52], [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*

$$\begin{aligned}\mathbb{E}\{\xi_{t+1}\mid G_t\}&\leq(1+\eta_t)\xi_t-u_t+c\theta_t\zeta_t+\mu_t\,,\\\mathbb{E}\{\zeta_{t+1}\mid G_t\}&\leq(1-\theta_t)\zeta_t-\bar{u}_t+\varepsilon_t\xi_t+\nu_t\end{aligned}\tag{70}$$

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:

$$\sum_{t=0}^{\infty}\eta_t<\infty\,,\sum_{t=0}^{\infty}\varepsilon_t<\infty\,,\sum_{t=0}^{\infty}\mu_t<\infty\,,\sum_{t=0}^{\infty}\nu_t<\infty\tag{71}$$

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

$$\sum_{t=0}^{\infty}u_t<\infty\,,\sum_{t=0}^{\infty}\bar{u}_t<\infty\,,\sum_{t=0}^{\infty}\theta_t\zeta_t<\infty\tag{72}$$

We construct coupled supermartingales that match the form of Lemma 2 using Lemma 1. First, use Lemma 1(ii)(45) as an upper bound on Lemma 1(iii) (46).

$$\begin{aligned}\mathbb{E}[J(Q_{t+1})\,|\,\mathcal{F}_t]\leq J(Q_t)-\alpha_t\left(1-\frac{\alpha_t G_Q^2}{\beta_t}\right)\|\nabla_Q J(Q_t)\|^2\\+\frac{\beta_t}{2}((1-\beta_t)(z_t-\bar{\delta}_{t-1})^2+\frac{L_Q}{\beta_t}\|Q_t-Q_{t-1}\|_{\mathcal{H}}^2\\+2\beta_t^2\sigma_\delta^2)+\frac{L_Q\sigma_Q^2\alpha_t^2}{2}+\varepsilon_t\|\nabla_Q J(Q_t)\|_{\mathcal{H}}\end{aligned}\tag{73}$$

We introduce three restrictions on the learning rate, expectation rate, and parsimony constant in order to simplify (73). 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 (73).

$$\begin{aligned}\mathbb{E}[J(Q_{t+1})\,|\,\mathcal{F}_t]\leq&J(Q_t)-\alpha_t\|\nabla_Q J(Q_t)\|^2\\&+\frac{\beta_t}{2}(z_t-\bar{\delta}_{t-1})^2+\frac{L_Q}{2}\|Q_t-Q_{t-1}\|_{\mathcal{H}}^2\\&+\beta_t^3\sigma_\delta^2+\alpha_t^2\left(\frac{L_Q\sigma_Q^2}{2}+\|\nabla_Q J(Q_t)\|_{\mathcal{H}}\right)\end{aligned}\tag{74}$$

Next, we aim to connect the result of (73) to the form of Lemma 2 via the identifications:

$$\xi_t = J(Q_t)\,, \zeta_t = (z_t - \bar{\delta}_{t-1})^2\,, \theta_t = \beta_t\,, c = 1/2 \tag{75}$$
$$u_t = \alpha_t \|\nabla_Q J(Q_t)\|^2, \eta_t = 0$$
$$\mu_t = \frac{L_Q}{2}\|Q_t - Q_{t-1}\|^2_{\mathcal{H}} + \beta_t^3 \sigma_\delta^2 + \alpha_t^2\left(\frac{L_Q \sigma_Q^2}{2} + \|\nabla_Q J(Q_t)\|_{\mathcal{H}}\right)$$

Observe that $\sum \mu_t < \infty$ due to the upper bound on $\|Q_t - Q_{t-1}\|^2_{\mathcal{H}}$ provided by Lemma 1(44) and the summability conditions for α_t^2 and β_t^2 (37).

Next, we identify terms in Lemma 1 (ii) (45) according to Lemma 2 in addition to (75).

$$\nu_t = \frac{L_Q}{\beta_t}\|Q_t - Q_{t-1}\|^2_{\mathcal{H}} + 2\beta_t^2 \sigma_\delta^2\,, \varepsilon_t = 0\,, \bar{u}_t = 0 \tag{76}$$

The summability of ν_t can be shown as follows: the expression $\|Q_t - Q_{t-1}\|^2_{\mathcal{H}}/\beta_t$ 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, $\sum_t \|Q_t - Q_{t-1}\|^2_{\mathcal{H}}/\beta_t < \infty$ almost surely. We also require $\sum_t \beta_t^2 < \infty$ (37) for the summability of the second term of (76)

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 nonnegative random variables w.p. 1, and that

$$\sum_t \alpha_t \|\nabla_Q J(Q_t)\|^2 < \infty, \quad \sum_t \beta_t (z_t - \bar{\delta}_{t-1})^2 < \infty \tag{77}$$

almost surely. Then, the summability of u_t taken together with non-summability of α_t and β_t (37) indicates that the limit infimum of the norm of the gradient of the cost goes to null.

$$\liminf_{t\to\infty} \|\nabla_Q J(Q_t)\| = 0\,, \liminf_{t\to\infty} (z_t - \bar{\delta}_{t-1})^2 = 0 \tag{78}$$

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 $\sum_{t=0}^{\infty} \|\alpha_t \nabla_Q J(Q_t)\|^2_{\mathcal{H}} \geq \sum_{t=t_0}^{\infty} \alpha_t \eta^2 = \infty$ which contradicts (77). Therefore, there exists a closed set $\bar{\mathcal{H}} \subset \mathcal{H}$ such that $\{Q_t\}$ visits $\bar{\mathcal{H}}$ infinitely often, and

$$\|\nabla_Q J(Q)\|_{\mathcal{H}} \begin{cases} \leq \eta \text{ for } Q \in \bar{\mathcal{H}} \\ > \eta \text{ for } Q \notin \bar{\mathcal{H}}, Q \in \{Q_t\} \end{cases} \tag{79}$$

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

$$\|\nabla_Q J(Q)\|_{\mathcal{H}} \begin{cases} \geq 2\eta \text{ for } Q \in \tilde{\mathcal{H}} \\ < 2\eta \text{ for } Q \notin \tilde{\mathcal{H}}, Q \in \{Q_t\} \end{cases} \tag{80}$$

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

$$\begin{aligned} &\|\nabla_Q J(Q_{k_i})\|_{\mathcal{H}} \geq 2\eta > \|\nabla_Q J(Q_t)\|_{\mathcal{H}} \\ &> \eta \geq \|\nabla_Q J(Q_{j_i})\|_{\mathcal{H}} \quad \text{for } t = k_i + 1, \ldots, j_i - 1 \end{aligned} \tag{81}$$

Therefore, we can write

$$\begin{aligned} \sum_{t\in\mathcal{T}} \|Q_{t+1} - Q_t\|_{\mathcal{H}} &= \sum_{i=1}^{\infty} \sum_{t=k_i}^{j_i-1} \|Q_{t+1} - Q_t\|_{\mathcal{H}} \\ &\geq \sum_{i=1}^{\infty} \|Q_{k_i} - Q_{j_i}\|_{\mathcal{H}} \geq \text{dist}(\bar{\mathcal{H}}, \tilde{\mathcal{H}}) = \infty \end{aligned} \tag{82}$$

However, we may also write that

$$\infty > \sum_{t=0}^{\infty} \alpha_t \|\nabla J(Q_t)\|^2_{\mathcal{H}} \geq \sum_{t\in\mathcal{T}} \alpha_t \|\nabla J(Q_t)\|^2_{\mathcal{H}} > \eta^2 \sum_{t\in\mathcal{T}} \alpha_t \tag{83}$$

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,

$$\sum_{t\in\mathcal{T}} \|Q_{t+1} - Q_t\|_{\mathcal{H}} \leq M \sum_{t\in\mathcal{T}} \alpha_t < \infty \tag{84}$$

which contradicts (82). 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 (77). Thus, any limit point of Q_t is a stationary point of $J(Q)$ almost surely. ■

Appendix C: Proof of Theorem 2

Begin with the expression in (46), and substitute in constant step-size selections with $\varepsilon = C\alpha^2$ and $(1-\beta) \leq 1$, and compute the total expectation ($\mathcal{F}_t = \mathcal{F}_0$) to write

$$\begin{aligned} \mathbb{E}[J(Q_{t+1})] \leq \mathbb{E}[J(Q_t)] &- \alpha\left(1 - \frac{\alpha G_Q^2}{\beta}\right)\mathbb{E}[\|\nabla_Q J(Q)\|^2] \\ &+ C\alpha^2 \mathbb{E}[\|\nabla_Q J(Q_t)\|_{\mathcal{H}}] + \frac{\beta}{2}\mathbb{E}[(z_{t+1} - \bar{\delta}_t)^2] \\ &+ \frac{L_Q \sigma_Q^2 \alpha^2}{2}\,, \end{aligned} \tag{85}$$

From here, we note that the sequence $\mathbb{E}[(z_{t+1} - \bar{\delta}_t)^2]$ is identical (except re-written in terms of Q-functions rather than value functions) to the sequence in Lemma 1(iii) of [27], and therefore, analogous reasoning regarding to that which yields eqn. (86) in Appendix D of [27] allows us to write

$$\mathbb{E}[(z_{t+1} - \bar{\delta}_t)^2] \leq \frac{2L_Q}{\beta^2}\left[\alpha^2(G_\delta^2 G_Q^2 + \lambda^2 D^2\right] + 2\beta\sigma_\delta^2\,, \tag{86}$$

which follows from applying (44) to the Hilbert-norm difference of Q-functions term and recursively substituting the total expectation of (45) back into itself, and simplifying the

resulting geometric sum. Now, we may substitute the right-hand side of (86) into the third term on the right-hand side of (85) to obtain

$$\begin{aligned}\mathbb{E}[J(Q_{t+1})] \leq \mathbb{E}[J(Q_t)] - \alpha\left(1-\frac{\alpha G_Q^2}{\beta}\right)\mathbb{E}[\|\nabla_Q J(Q)\|^2] \\ + C\alpha^2\mathbb{E}[\|\nabla_Q J(Q_t)\|_{\mathcal{H}}] + \frac{2L_Q}{\beta}\left[\alpha^2(G_\delta^2 G_Q^2 + \lambda^2 D^2\right] \\ + 2\beta^2\sigma_\delta^2 + \frac{L_Q\sigma_Q^2\alpha^2}{2}, \end{aligned} \tag{87}$$

The rest of the proof proceeds as follows: we break the right-hand side of (87) into two subsequences, one in which the expected norm of the cost functional's gradient $\mathbb{E}[\|\nabla_Q J(Q_t)\|_{\mathcal{H}}]$ is below a specified threshold, whereby $J(Q_t)$ is a decreasing sequence in expectation, and one where this condition is violated. We can use this threshold condition to define a deterministically decreasing auxiliary sequence to which the Monotone Convergence Theorem applies, and hence we obtain convergence of the auxiliary sequence. Consequently, we obtain convergence in infimum of the expected value of $J(Q_t)$ to a neighborhood.

We proceed by defining the conditions under which $\mathbb{E}[J(Q_t)]$ is decreasing, i.e.,

$$\begin{aligned}\mathbb{E}[J(Q_{t+1})] \leq \mathbb{E}[J(Q_t)] - \alpha\left(1-\frac{\alpha G_Q^2}{\beta}\right)\mathbb{E}[\|\nabla_Q J(Q)\|^2] \\ + C\alpha^2\mathbb{E}[\|\nabla_Q J(Q_t)\|_{\mathcal{H}}] + \frac{2L_Q}{\beta}\left[\alpha^2(G_\delta^2 G_Q^2 + \lambda^2 D^2\right] \\ + 2\beta^2\sigma_\delta^2 + \frac{L_Q\sigma_Q^2\alpha^2}{2} \\ \leq \mathbb{E}[J(Q_t)] \end{aligned} \tag{88}$$

Note that (88) holds whenever the following is true:

$$\begin{aligned} -\alpha\left(1-\frac{\alpha G_Q^2}{\beta}\right)\mathbb{E}[\|\nabla_Q J(Q)\|^2] + C\alpha^2\mathbb{E}[\|\nabla_Q J(Q_t)\|_{\mathcal{H}}] \\ + \frac{2L_Q}{\beta}\left[\alpha^2(G_\delta^2 G_Q^2 + \lambda^2 D^2\right] + 2\beta^2\sigma_\delta^2 + \frac{L_Q\sigma_Q^2\alpha^2}{2} \leq 0 \end{aligned} \tag{89}$$

Observe that the left-hand side of (89) defines a quadratic function of $\mathbb{E}[\|\nabla_Q J(Q_t)\|_{\mathcal{H}}]$ which opens downward. We can solve for the condition under which (89) holds with equality by obtaining the positive root (since $\mathbb{E}[\|\nabla_Q J(Q_t)\|_{\mathcal{H}}] \geq 0$) of this expression through the quadratic formula:

$$\begin{aligned}\mathbb{E}[\|\nabla_Q J(Q)\|] &= \left(C + \left[C^2 + 4\left(\frac{1}{\alpha} - \frac{G_Q^2}{\beta}\right)\left[\frac{2L_Q}{\beta}[(G_\delta^2 G_Q^2 + \lambda^2 D^2)]\right.\right.\right. \\ &\quad \left.\left.\left. + \frac{2\beta^2\sigma_\delta^2}{\alpha^2} + \frac{L_Q\sigma_Q^2}{2}\right]\right]^{1/2}\right)\left(\frac{1}{\alpha} - \frac{G_Q^2}{\beta}\right)^{-1} \\ &= \mathcal{O}\left(\frac{\alpha\beta}{\beta-\alpha}\left[1 + \sqrt{1 + \frac{\beta-\alpha}{\alpha\beta}\left(\frac{1}{\beta} + \frac{\beta^2}{\alpha^2}\right)}\right]\right) \end{aligned} \tag{90}$$

where we have cancelled out a common factor of α^2 as well as common factors of -1 throughout. Now, define the right-hand side of (90) as a constant Δ, and the auxiliary sequence

$$\begin{aligned}\Gamma_t = \mathbb{E}[J(Q_t)]\mathbb{1}\Big\{ \min_{u\leq t} -\alpha\left(1-\frac{\alpha G_Q^2}{\beta}\right)\mathbb{E}[\|\nabla_Q J(Q)\|^2] \\ + C\alpha^2\mathbb{E}[\|\nabla_Q J(Q_t)\|_{\mathcal{H}}] + \frac{2L_Q}{\beta}\left[\alpha^2(G_\delta^2 G_Q^2 \right. \\ \left. + \lambda^2 D^2\right] + 2\beta^2\sigma_\delta^2 + \frac{L_Q\sigma_Q^2\alpha^2}{2} > \Delta\Big\} \end{aligned} \tag{91}$$

where $\mathbb{1}\{E\}$ denotes the indicator function of a (deterministic) event E. From here, we note that Γ_t is nonnegative since $J(Q_t) \geq 0$. Moreover, Γ_t is decreasing: either the indicator is positive, in which case its argument is true, and hence (89) holds with equality. When (89) holds with equality, the objective is decreasing, namely, (88) is valid. Alternatively, condition inside the indicator is null, which, due to the use of the minimum in the definition (91), means that the indicator is null for all subsequent times. Therefore, in either case, Γ_t is nonnegative and decreasing, and therefore we may apply the Monotone Convergence Theorem [53] to conclude $\Gamma_t \to 0$. Therefore, we have either that $\lim_t \mathbb{E}[J(Q_t)] - \Delta = 0$ or that the indicator in (91) is null for $t \to \infty$. Taken together, these statements allow us to conclude

$$\liminf_{t\to\infty} \mathbb{E}[J(Q_t)] \leq \mathcal{O}\left(\frac{\alpha\beta}{\beta-\alpha}\left[1 + \sqrt{1 + \frac{\beta-\alpha}{\alpha\beta}\left(\frac{1}{\beta} + \frac{\beta^2}{\alpha^2}\right)}\right]\right) \tag{92}$$

which is as stated in (38). ■