In *IEEE Transactions on Automatic Control*

Policy Evaluation in Continuous MDPs with Efficient Kernelized Gradient Temporal Difference

Alec Koppel, *Member, IEEE*, Garrett Warnell, Ethan Stump, Peter Stone, and Alejandro Ribeiro *Member, IEEE*

***Abstract*—We consider policy evaluation in infinite-horizon discounted Markov decision problems (MDPs) with continuous compact state and action spaces. We reformulate this task as a compositional stochastic program with a function-valued decision variable that belongs to a reproducing kernel Hilbert space (RKHS). We approach this problem via a new functional generalization of stochastic quasi-gradient methods operating in tandem with stochastic sparse subspace projections. The result is an extension of gradient temporal difference learning that yields nonlinearly parameterized value function estimates of the solution to the Bellman evaluation equation. We call this method Parsimonious Kernel Gradient Temporal Difference (PKGTD) Learning. Our main contribution is a memory-efficient non-parametric stochastic method guaranteed to converge exactly to the Bellman fixed point with probability 1 with attenuating step-sizes under the hypothesis that it belongs to the RKHS. Further, with constant step-sizes and compression budget, we establish mean convergence to a neighborhood and that the value function estimates have finite complexity. In the Mountain Car domain, we observe faster convergence to lower Bellman error solutions than existing approaches with a fraction of the required memory.**

I. Markov Decision Processes

We consider an autonomous agent acting in an environment defined by a Markov decision process (MDP) [1] with continuous spaces, which is increasingly relevant to emerging technologies such as robotics [2], power systems [3], and others. A MDP is a quintuple $(\mathcal{X}, \mathcal{A}, \mathbb{P}, r, \gamma)$, where $\mathbb{P}$ is the action-dependent transition probability of the process: when the agent starts in state $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^p$ at time t and takes an action $\mathbf{a}_t \in \mathcal{A}$, a transition to next state $\mathbf{y}_t \in \mathcal{X}$ is distributed according to $\mathbf{y}_t \sim \mathbb{P}(\cdot \,|\, \mathbf{x}_t, \mathbf{a}_t)$. After transitioning to a particular $\mathbf{y}_t$, the MDP reveals an instantaneous reward $r(\mathbf{x}_t, \mathbf{a}_t, \mathbf{y}_t)$, where the reward function is a map $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$.

We focus on *policy evaluation*: control decisions $\mathbf{a}_t$ are chosen according to a fixed stationary stochastic policy $\pi : \mathcal{X} \to \rho(\mathcal{A})$, where $\rho(\mathcal{A})$ denotes the set of probability distributions over $\mathcal{A}$. Policy evaluation underlies methods that seek optimal policies through repeated evaluation and improvement. In policy evaluation, we seek to compute the *value* of a policy when starting in state $\mathbf{x}$, quantified by the discounted expected sum of rewards, or value function $V^\pi(\mathbf{x})$:[1]

$$V^\pi(\mathbf{x}) = \mathbb{E}_{\mathbf{y}}\Big[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{x}_t, \mathbf{a}_t, \mathbf{y}_t) \big| \mathbf{x}_0 = \mathbf{x}, \{\mathbf{a}_t = \pi(\mathbf{x}_t)\}_{t=0}^{\infty}\Big]. \quad (1)$$

For a single trajectory through the state space $\mathcal{X}$, $\mathbf{y}_t = \mathbf{x}_{t+1}$. The value function (1) is parameterized by a discount factor $\gamma \in (0, 1)$ that determines farsightedness. Decomposing the summand in (1) into its first and subsequent terms, and using both the stationarity of the transition probability and the Markov property yields the Bellman evaluation equation [4]:

$$V^\pi(\mathbf{x}) = \int_{\mathcal{X}} [r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V^\pi(\mathbf{y})] \mathbb{P}(d\mathbf{y} \,|\, \mathbf{x}, \pi(\mathbf{x})) \quad (2)$$

for all $\mathbf{x} \in \mathcal{X}$. The right-hand side of (2) defines a Bellman evaluation operator $\mathscr{B}^\pi : \mathcal{B}(\mathcal{X}) \to \mathcal{B}(\mathcal{X})$ over $\mathcal{B}(\mathcal{X})$, the space of bounded continuous value functions $V : \mathcal{X} \to \mathbb{R}$:

$$(\mathscr{B}^\pi V)(\mathbf{x}) = \int_{\mathcal{X}} [r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V(\mathbf{y})] \mathbb{P}(d\mathbf{y} \,|\, \mathbf{x}, \pi(\mathbf{x})) \quad (3)$$

for all $\mathbf{x} \in \mathcal{X}$. Proposition 4.2(b) in [5] establishes that the stationary point of (3) is V^π, i.e., $(\mathscr{B}^\pi V^\pi)(\mathbf{x}) = V^\pi(\mathbf{x})$. As a stepping stone to finding optimal policies in infinite MDPs, we seek here to find the fixed point of (3). Specifically, the goal of this work is stable value function estimation in infinite MDPs, with nonlinear parameterizations that are allowed to be infinite, but are nonetheless memory-efficient.

Challenges To solve (3), fixed point methods, i.e., value iteration ($V_{k+1} = \mathscr{B}^\pi V_k$), have been proposed [5], but can only be implemented in a memory-affordable manner when the value function can be represented by a vector whose length is defined by the number of states and the state space is small enough that the expectation[2] in $\mathscr{B}$ can be computed. For large spaces, stochastic approximations of value iteration, i.e., temporal difference (TD) learning [6], circumvent computing expectations. Incremental methods (least-squares TD) are an alternative when $V(\mathbf{x})$ is vector-valued [7], but extensions to infinite representations require infinite memory [8].

Solving the fixed point problem defined by (3) requires surmounting the fact that this expression is defined for each $\mathbf{x} \in \mathcal{X}$, which for continuous $\mathcal{X} \subset \mathbb{R}^p$ has *infinitely many* unknowns. This phenomenon is one example of Bellman's curse of dimensionality [4], and it is frequently sidestepped

This manuscript was originally submitted for review on December 6, 2017, and supported by the SMART Scholarship, ARL DCIST CRA W911NF-17-2-0181, NSF DGE-1321851, Intel DevCloud and Intel Science and Technology Center for Wireless Autonomous Systems (ISTC-WAS).

[1] Computational and Information Sciences Directorate, U.S. Army Research Laboratory, 2800 Powder Mill Rd., Adelphi, MD 20783, email: {alec.e.koppel,garrett.a.warnell,ethan.a.stump2}.civ@mail.mil.

[2] Department of Computer Science, University of Texas at Austin, 2317 Speedway, Stop D9500 Austin, Texas 78712-1757 USA, email: pstone@cs.utexas.edu,

[3] Department of Electrical and Systems Engineering, University of Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104

[1] In MDPs more generally, we choose actions $\{\mathbf{a}_t\}_{t=1}^{\infty}$ to maximize the reward accumulation starting from state $\mathbf{x}$, i.e., $V(\mathbf{x}, \{\mathbf{a}_t\}_{t=0}^{\infty}) = \mathbb{E}_{\mathbf{y}}\Big[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{x}_t, \mathbf{a}_t, \mathbf{y}_t) \,\big|\, \mathbf{x}_0 = \mathbf{x}, \{\mathbf{a}_t\}_{t=0}^{\infty}\Big]$. For a fixed policy π, the setting of this work, this simplifies to (1).

[2] Observe that the integral in (2) defines a conditional expectation: $V^\pi(\mathbf{x}) = \mathbb{E}_{\mathbf{y}}[r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V^\pi(\mathbf{y})] \,|\, \mathbf{x}, \pi(\mathbf{x})]$.

by parameterizing the value function using a finite linear [9], [10] or nonlinear [11] basis expansion. Such methods have paved the way for the recent success of neural networks in value function-based approaches to MDPs, but combining TD learning with different parameterizations may cause divergence [12]: in general, the representation must be tied to the stochastic update [13] to ensure both are stable.

Contributions Our main result is a memory-efficient, nonparametric, stochastic method that converges to the Bellman fixed point almost surely when it belongs to a reproducing kernel Hilbert space (RKHS). The hypothesis that the value function belongs to a RKHS restricts the relationship between rewards and value to be smooth (see Assumption 4 [cf. (34)]), i.e., large changes in rewards yield large changes in value., which holds, for instance, when the reward is a potential or navigation function [14], [15]. Such specifications are known to interact favorably in controller design from a dynamical systems perspective. We reformulate (2) as a compositional stochastic program (Section II), a topic studied in operations research [16] and probability [17]. These problems motivate stochastic *quasi-gradient* (SQG) methods, i.e., two time-scale approaches, to mitigate the fact that the objective's stochastic gradient requires evaluating an expectation [18].

Two time-scale stochastic methods have a significant history in reinforcement learning in the context of a class of policy learning algorithms called actor-critic [19], which mix together gradient ascent on the value function [20] with value function estimation through temporal differences [6]. Alternatively, their utility also has been established for policy evaluation in [11], [21] for finite MDPs or value functions that have finite vector-valued parameterizations. Our work is more closely related to this later context; however, a key point of departure is that we propose to operate directly in a function space, motivated by the fact that policy evaluation in a continuous MDP defines a functional fixed point problem. Specifically, we use SQG in infinite MDPs.

In (2), the decision variable is a continuous function, which we address by hypothesizing the Bellman fixed point belongs to a RKHS [22]. However, a function in a RKHS has comparable complexity to the number of training samples processed, which could be infinite (an issue ignored in many kernel methods for MDPs [23]–[27]). Specifically, it's well known that for a kernelized interpolator defined by a training set of N samples, the complexity is $\mathcal{O}(N)$. Thus, to solve the population problem defined by Bellman's equations, $N \to \infty$, and thus so is the complexity of the function estimate. We propose to tackle this memory bottleneck by requiring memory efficiency in both the function sample path and in its limit, whose complexity in the worst case is defined by the metric entropy of the state space (Corollary 1).

To find a memory-efficient sample path in the function space, we generalize SQG to RKHSs (Section III), and combine this generalization with greedily-constructed sparse subspace projections (Section III-A). These subspaces are constructed via matching pursuit [28], [29], a procedure motivated by the facts that kernel matrices induced by arbitrary data streams likely violate requirements for convex-relaxation-based sparsity [30]. Rather than unsupervised forgetting [31], we tie the projection-induced error to ensure the stochastic gradient still satisfies a descent property [32], thus keeping only those dictionary points needed to converge (Section IV).

Whereas in [32], compressed kernel methods are analyzed for supervised learning and a tunable tradeoff between memory and sub-optimality is provided, here we study compressed kernel methods in the context of policy evaluation in reinforcement learning. This later context requires surmounting the technical challenges associated with nested expectations, specifically, that SQG is defined by coupled supermartingales, rather than standard stochastic descent arguments in [32], and hence has fundamental qualitative and quantitative departures from existing works on RKHS learning.

We note that this hard-thresholding projection could be applied to other stochastic algorithms in RKHS for reinforcement learning such as [23]–[25], [27], but applying them to incremental methods (LSTD) [8], [26] remains elusive since relating the per-step bias caused by sparsification to ensure valid descent directions is elusive.

As a result, we conduct functional SQG descent via sparse projections of the SQG. This maintains a moderate-complexity sample path exactly towards V^*, which may be made arbitrarily close to the Bellman fixed point by decreasing the regularizer. By generalizing the relationship between SQG and supermartingales in [33] to Hilbert spaces, we establish that the sparse projected SQG sequence converges almost surely to the Bellman fixed point with decreasing learning rates, and converges in mean while maintaining finite complexity when constant learning rates are used (Section IV). We then empirically evaluate the proposed value function approximation method on the discrete Mountain Car domain in Section V and summarize our findings in Section VI.

We would like to point out that convergence of two time-scale methods is well-understood [33], [34]; however, applying these methods as is requires decision variables to be *vectors*, not *functions*. This parameterization, however, causes an approximation error which is difficult to characterize (see [26]). In contrast, the RKHS parameterization, operating with the combination of projections and SQG, attains solutions that are close to the minimizer of the true Bellman evaluation error, where closeness is controlled by regularization introduced in the next section. This is due to the fact that RKHS possesses universal approximation under judicious choice of the kernel [35], thus circumventing approximation error.

Recently, in companion work, an optimization-based variant of Q learning in RKHS is developed [36]; however, a number of essential points distinguish that thread from methods developed here. Specifically, the optimization-based reformulation of Bellman's evaluation equation yields a convex program for which i.i.d. assumptions are close-to-valid. By contrast, Bellman's optimality equation yields a *non-convex reformulation*. While the convergence of Q learning requires i.i.d. assumptions, typically in practice these are violated. These statistical dependencies make policy learning a challenging domain to study Bayesian exploration, whereas policy evaluation is suitable [37]. Additionally, policy evaluation is just one component of reinforcement learning algorithms based upon policy search such as policy gradient method [20] or actor-critic [38], [39], whereas Q learning is a standalone procedure. Other recent work focuses on policy search, a form of stochastic gradient with respect to a parameterized family of policies [40], [41], which is categorically different from fixed point iterations derived from Bellman's equations [4].

II. Policy Evaluation

We reformulate the fixed point problem (3) defined by Bellman's equation so that it may be identified with a nested stochastic program. Since the resulting domain of this problem is intractable, we hypothesize that the Bellman fixed point belongs to a RKHS. Then, to apply the Representer Theorem, we require the introduction of regularization.

We proceed with reformulating (3): subtract the value function $V^\pi(\mathbf{x})$ that satisfies the fixed point relation from both sides, and then pull it inside the expectation:

$$0 = \mathbb{E}_{\mathbf{y}}[r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V^\pi(\mathbf{y}) - V^\pi(\mathbf{x}) \,\big|\, \mathbf{x}, \pi(\mathbf{x})] \tag{4}$$

for all $\mathbf{x} \in \mathcal{X}$. Value functions satisfying (4) are equivalent to those which satisfy the quadratic expression $0 = \frac{1}{2}(\mathbb{E}_{\mathbf{y}}[r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V^\pi(\mathbf{y}) - V^\pi(\mathbf{x}) \,\big|\, \mathbf{x}, \pi(\mathbf{x})])^2$, which is null for all $\mathbf{x} \in \mathcal{X}$, the starting point of the trajectory defining the value function (1). To solve this expression for every $\mathbf{x}$, rather than solving it for a fixed $\mathbf{x}$ separately, we may integrate it out, which we do together with integrating over policy $\pi(\mathbf{x})$ to yield the compositional stochastic program:

$$\begin{aligned} V^\pi &= \operatorname*{argmin}_{V \in \mathcal{B}(\mathcal{X})} J(V) \qquad (5)\\ &:= \operatorname*{argmin}_{V \in \mathcal{B}(\mathcal{X})} \mathbb{E}_{\mathbf{x},\pi(\mathbf{x})}\Big\{\frac{1}{2}(\mathbb{E}_{\mathbf{y}}[r(\mathbf{x},\pi(\mathbf{x}),\mathbf{y}) + \gamma V(\mathbf{y}) - V(\mathbf{x}) \big| \mathbf{x}, \pi(\mathbf{x})])^2\Big\} \end{aligned}$$

whose solutions coincide exactly with the fixed points of (3). The equivalence of (4) and (5) is not in general true, but only true when the probability distribution μ over $\mathbf{x}$ is ergodic. That is, for fixed policy π, μ is non-vanishing over the entire state space $\mathcal{X}$: for each $\mathbf{x} \in \mathcal{X}$, $\mu(\mathbf{x}) > 0$. Henceforth, we require μ, the prior distribution over states $\mathbf{x} \in \mathcal{X}$, to be ergodic. See [11], [42] for a discussions of transforming Bellman equations into objective functions, and the necessity of ergodicity.

(5) defines a functional optimization problem which is intractable when we search over all bounded continuous functions $\mathcal{B}(\mathcal{X})$. However, when we restrict $\mathcal{B}(\mathcal{X})$ to a Hilbert space $\mathcal{H}$ equipped with a unique *reproducing kernel*, i.e., an inner product-like map $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for $f \in \mathcal{H}$,

$$(i)\ \langle f, \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = f(\mathbf{x}) \ \ (ii)\ \mathcal{H} = \overline{\text{span}\{\kappa(\mathbf{x}, \cdot)\}}\ , \tag{6}$$

for all $\mathbf{x} \in \mathcal{X}$. We may apply the Representer Theorem to transform (5) into a semi-parametric one [22], [43]. In a RKHS, the optimal function $f \in \mathcal{H}$ of (5) then takes the form

$$f(\mathbf{x}) = \sum_{n=1}^{N} w_n \kappa(\mathbf{x}_n, \mathbf{x})\ , \tag{7}$$

where $\mathbf{x}_n$ is a realization of the random variable $\mathbf{x}$. Thus, $f \in \mathcal{H}$ is a kernel expansion *only* at training samples. We define the upper summand index N in (7) in the kernel expansion of $f \in \mathcal{H}$ as the model order, which here coincides with the training sample size. Common kernel choices are polynomials and radial basis (Gaussian) functions, i.e., $\kappa(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T\mathbf{x}' + b\right)^c$ and $\kappa(\mathbf{x}, \mathbf{x}') = \exp\{-\|\mathbf{x} - \mathbf{x}'\|_2^2/2c^2\}$, respectively. In (6), property (i) is called the reproducing property, which follows from Riesz Representation Theorem [44]. Replacing f by $\kappa(\mathbf{x}', \cdot)$ in (6) (i) yields $\langle \kappa(\mathbf{x}', \cdot), \kappa(\mathbf{x}, \cdot)\rangle_{\mathcal{H}} = \kappa(\mathbf{x}, \mathbf{x}')$, the origin of the term "reproducing kernel." Moreover, property (6) (ii) states that functions $f \in \mathcal{H}$ admit a basis expansion in terms of kernels (7). Such spaces are called reproducing kernel Hilbert spaces (RKHSs). When the kernel is universal [35], e.g., a Gaussian, a continuous function over a compact set may be approximated uniformly by one in a RKHS.

Subsequently, we seek to solve (5) with $V \in \mathcal{H}$, and independent and identically distributed samples $(\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ from the triple $(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y})$ are sequentially available, yielding

$$\begin{aligned} V^* = \operatorname*{argmin}_{V \in \mathcal{H}} \mathbb{E}_{\mathbf{x},\pi(\mathbf{x})}\Big\{\frac{1}{2}(\mathbb{E}_{\mathbf{y}}[r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) \qquad (8)\\ + V(\mathbf{y}) - V(\mathbf{x}) \,\big|\, \mathbf{x}, \pi(\mathbf{x})])^2\Big\} + \frac{\lambda}{2}\|V\|_{\mathcal{H}}^2 \end{aligned}$$

Hereafter, define $L(V) := \mathbb{E}_{\mathbf{x},\pi(\mathbf{x})}\{\frac{1}{2}(\mathbb{E}_{\mathbf{y}}[r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V(\mathbf{y}) - V(\mathbf{x}) \,\big|\, \mathbf{x}, \pi(\mathbf{x})])^2\}$ and $J(V) = L(V) + (\lambda/2)\|V\|_{\mathcal{H}}^2$. The regularization term $(\lambda/2)\|V\|_{\mathcal{H}}^2$ in (8) is needed to apply the Representer Theorem (7) [22]. Thus, policy evaluation in infinite MDPs (8) is both a specialization of compositional stochastic programming [33] to an objective defined by dynamic programming, and a generalization to the case where the decision variable is not vector-valued but is instead a function.

III. Functional Stochastic Quasi-Gradient

To apply functional SQG to (8), we differentiate the compositional objective $L(V)$, which is of the form $L = g \circ h$, with $g(u) = \mathbb{E}_{\mathbf{x},\pi(\mathbf{x})}[(1/2)u^2]$ and $h(V) = \mathbb{E}_{\mathbf{y}}[r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V(\mathbf{y}) - V(\mathbf{x}) \,\big|\, \mathbf{x}, \pi(\mathbf{x})]$, and then consider its stochastic estimate. Consider the Frechét derivative of $L(V)$:

$$\begin{aligned} \nabla_V L(V) = \mathbb{E}_{\mathbf{x},\pi(\mathbf{x})}\big\{\mathbb{E}_{\mathbf{y}}[\gamma\kappa(\mathbf{y}, \cdot) - \kappa(\mathbf{x}, \cdot) \,\big|\, \mathbf{x}, \pi(\mathbf{x})] \qquad (9)\\ \times \mathbb{E}_{\mathbf{y}}[r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V(\mathbf{y}) - V(\mathbf{x}) \big| \mathbf{x}, \pi(\mathbf{x})]\big\} \end{aligned}$$

Here we pull the differential operator inside the expectation and use both the chain rule and reproducing property of the kernel (6)(i). For future reference, we define the expression $\mathbb{E}_{\mathbf{y}}[r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V(\mathbf{y}) - V(\mathbf{x}) \,\big|\, \mathbf{x}, \pi(\mathbf{x})] = \bar{\delta}$ as the average temporal difference [6]. To perform stochastic descent in function space $\mathcal{H}$, we need a stochastic approximate of (9) evaluated at a state-action-state triple $(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y})$, which together with the regularizer yields

$$\begin{aligned} &\nabla_V J(V, \delta; \mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) \qquad (10)\\ &\quad = [\gamma\kappa(\mathbf{y}, \cdot) - \kappa(\mathbf{x}, \cdot)][r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V(\mathbf{y}) - V(\mathbf{x})] + \lambda V \end{aligned}$$

where $\delta := r(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y}) + \gamma V(\mathbf{y}) - V(\mathbf{x})$ is defined as the (instantaneous) temporal difference. Observe that we cannot obtain samples of $\nabla_V J(V, \delta; \mathbf{x}, \pi(\mathbf{x}), \mathbf{y})$ with a single query to a simulation oracle: stochastic gradient method would estimate one of the expected gradients by its instantaneous approximation, but would still leave a second expected value that depends on infinitely many realizations of either prior distribution and policy $(\mathbf{x}, \pi(\mathbf{x}))$ or MDP transition dynamics $\mathbf{y}$, a problem first identified in [21] for finite MDPs where it is called the *double sampling problem.* Double sampling aside, an equally significant challenge associated with using (10) as a candidate descent direction is that classically we would compute its expectation conditional on the algorithm history, but due to the dependence of the factors, this does not yield (9). In particular, the stochastic gradient is *biased* with respect to (9) due to the inner conditional expectation in (9).

To mitigate these issues, we require a method that constructs a *coupled* stochastic descent procedure by considering noisy estimates of both factors in the product-of-expectations expression in (9). The first factor $[\gamma\kappa(\mathbf{y},\cdot)-\kappa(\mathbf{x},\cdot)]$ in (10) is a difference of kernel maps, so estimating its expectation is parameterized by infinitely many samples of $\mathbf{x}$ and $\mathbf{y}$ [45], [46]. Instead, we propose a sequence based on samples of the second scalar factor to estimate its expected value. Specifically, from samples of δ, consider a recursion z_t that estimates $\bar{\delta}$ as

$$\begin{aligned}\delta_t &= r(\mathbf{x}_t,\pi(\mathbf{x}_t),\mathbf{y}_t)+\gamma V_t(\mathbf{y}_t)-V_t(\mathbf{x}_t)\\ z_{t+1} &= (1-\beta_t)z_t+\beta_t\delta_t \end{aligned} \tag{11}$$

where we define δ_t [6] as the temporal difference at time t in (11). Thus, (11) averages the TD sequence δ_t: z_t estimates $\bar{\delta}_t$, and $\beta_t \in (0,1)$ is a learning rate.

To define a stochastic descent step, we replace the first factor inside the outer expectation in (9) with its instantaneous approximate, i.e., $[\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot)]$, at sample $(\mathbf{x}_t,\pi(\mathbf{x}_t),\mathbf{y}_t)$, which yields the stochastic quasi-gradient step

$$\hat{V}_{t+1} = (1-\alpha_t\lambda)\hat{V}_t-\alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))z_{t+1}\;. \tag{12}$$

where the coefficient $(1-\alpha_t\lambda)$ comes from the regularizer, and α_t is a positive scalar learning rate. This update is a stochastic quasi-gradient step because the true stochastic gradient of $J(V)$ is $(\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))\delta_t$, but this estimator is unavailable with a single trajectory of the MDP since the factors in this product are dependent. By replacing δ_t by auxiliary variable z_{t+1} this issue may be circumvented in the construction of coupled supermartingales (Section IV).

Kernel Parameterization Suppose $V_0 = 0 \in \mathcal{H}$. Then, (12) at time t, making use of the Representer Theorem (7), implies the function $\tilde{V}_t$ is a kernel expansion of past states $(\mathbf{x}_t,\mathbf{y}_t)$ as

$$\hat{V}_t(\mathbf{x}) = \sum_{n=1}^{2(t-1)} w_n\kappa(\mathbf{v}_n,\mathbf{x}) = \mathbf{w}_t^T\boldsymbol{\kappa}_{\mathbf{X}_t}(\mathbf{x})\;. \tag{13}$$

On the right-hand side of (13) we introduce the notation $\mathbf{v}_n = \mathbf{x}_{n/2}$ for n even and $\mathbf{v}_n = \mathbf{y}_{n/2+1}$ for n odd, and: $\mathbf{w}_t = [w_1,\cdots,w_{2(t-1)}] \in \mathbb{R}^{2(t-1)}$, $\mathbf{X}_t = [\mathbf{x}_1,\mathbf{y}_1,\ldots,\mathbf{x}_{t-1},\mathbf{y}_{t-1}] \in \mathbb{R}^{p\times 2(t-1)}$, and $\boldsymbol{\kappa}_{\mathbf{X}_t}(\cdot) = [\kappa(\mathbf{x}_1,\cdot),\ \kappa(\mathbf{y}_1,\cdot),\ \ldots\ ,\kappa(\mathbf{x}_{t-1},\cdot),\kappa(\mathbf{y}_{t-1},\cdot)]^T$. The kernel expansion in (13), together with the functional update (12), yields the fact that functional SQG in $\mathcal{H}$ amounts to the following updates on the data matrix $\mathbf{X}$, henceforth referred to as a kernel dictionary, and coefficient vector $\mathbf{w}$:

$$\begin{aligned}&\mathbf{X}_{t+1} = [\mathbf{X}_t\ ,\ \mathbf{x}_t\ ,\ \mathbf{y}_t],\\ &\mathbf{w}_{t+1} = [(1-\alpha_t\lambda)\mathbf{w}_t\ ,\ \alpha_t z_{t+1}\ ,\ -\alpha_t\gamma z_{t+1}]\;,\end{aligned} \tag{14}$$

Observe that this update causes $\mathbf{X}_{t+1}$ to have two more columns than $\mathbf{X}_t$. We define the *model order* as number of data points 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 $\hat{V}_t(\mathbf{x})$ is infinite.

A. Sparse Stochastic Subspace Projections

Since the update (12) has complexity $\mathcal{O}(t)$ due to the RKHS parameterization [32], [45], it is impractical in settings with streaming data or arbitrarily large training sets. We address this issue by replacing the stochastic descent step (12) with an orthogonally projected variant [32], where the projection is onto a low-dimensional functional subspace $\mathcal{H}_{\mathbf{D}_{t+1}}$ of $\mathcal{H}$, i.e.,

$$V_{t+1}=\mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}[(1-\alpha_t\lambda)V_t-\alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))z_{t+1}], \tag{15}$$

where α_t again is a scalar step-size, and $\mathcal{H}_{\mathbf{D}_{t+1}} = \text{span}\{\kappa(\mathbf{d}_n,\cdot)\}_{n=1}^{M_t}$ for some collection of sample instances $\{\mathbf{d}_n\} \subset \{\mathbf{x}_u\}_{u\le t}$. Note that the un-projected function SQG method (12) may be interpreted as conducting a sequence of orthogonal projections, which motivates the design of (15). Specifically, rewrite (12) as the quadratic minimization

$$\begin{aligned}\hat{V}_{t+1}&=\underset{V\in\mathcal{H}}{\text{argmin}}\left\|V-\Big((1-\alpha_t\lambda)\hat{V}_t-\alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))z_{t+1}\Big)\right\|_{\mathcal{H}}^2\\ &=\underset{V\in\mathcal{H}_{\mathbf{X}_{t+1}}}{\text{argmin}}\left\|V-\Big((1-\alpha_t\lambda)\hat{V}_t-\alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))z_{t+1}\Big)\right\|_{\mathcal{H}}^2,\end{aligned} \tag{16}$$

where the first equality in (16) comes from ignoring constant terms which vanish upon differentiation with respect to V, and the second comes from observing that V_{t+1} can be represented using only the points $\mathbf{X}_{t+1}$, using (14). Notice (16) expresses V_{t+1} as the projection $(1-\alpha_t\lambda)V_t-\alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))z_{t+1}$ onto the subspace defined by dictionary $\mathbf{X}_{t+1}$.

Rather than selecting dictionary $\mathbf{D} = \mathbf{X}_{t+1}$, we propose instead to select a different dictionary, $\mathbf{D} = \mathbf{D}_{t+1}$, which is extracted from the data points observed thus far, at each iteration. The process by which we select $\mathbf{D}_{t+1}$ is delayed for now, but is of dimension $p \times M_{t+1}$. We design a scheme such that M_{t+1} is *independent* of t, and instead determined by fundamental topological properties of state space $\mathcal{X}$, i.e., a generalization of the Nyquist rate [47]. As a result, the sequence V_t differs from the functional stochastic quasi-gradient method $\hat{V}_t$ presented at the outset of this section.

Specifically, suppose the function V_{t+1} is parameterized dictionary $\mathbf{D}_{t+1}$ and weight vector $\mathbf{w}_{t+1}$. We denote columns of $\mathbf{D}_{t+1}$ as $\mathbf{d}_n$ for $n = 1,\ldots,M_{t+1}$, where the time index is dropped for notational clarity but may be inferred from the context. Setting aside how $\mathbf{D}_{t+1}$ is chosen for now, we replace the update (16) in which the dictionary grows at each iteration by the functional stochastic quasi-gradient sequence projected onto the subspace $\mathcal{H}_{\mathbf{D}_{t+1}} = \text{span}\{\kappa(\mathbf{d}_n,\cdot)\}_{n=1}^{M_{t+1}}$ as

$$\begin{aligned}V_{t+1}&=\underset{V\in\mathcal{H}_{\mathbf{D}_{t+1}}}{\text{argmin}}\left\|V-\Big((1-\alpha_t\lambda)\hat{V}_t-\alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))z_{t+1}\Big)\right\|_{\mathcal{H}}^2,\\ &:=\mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}\Big[(1-\alpha_t\lambda)V_t-\alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))z_{t+1}\Big].\end{aligned} \tag{17}$$

where we define the projection operator $\mathcal{P}$ onto subspace $\mathcal{H}_{\mathbf{D}_{t+1}} \subset \mathcal{H}$ by the update (17). This orthogonal projection is the modification of the functional SQG iterate [cf. (12)]

Algorithm 1 PKGTD: Parsimonious Kernel Gradient Temporal Difference

```
Require: $\{\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t, \alpha_t, \beta_t, \epsilon_t\}_{t=0,1,2,\ldots}$
  initialize $V_0(\cdot) = 0, \mathbf{D}_0 = [], \mathbf{w}_0 = [], z_0 = 0$, i.e. initial dict., coeffs., and aux. variable null
  for $t = 0, 1, 2, \ldots$ do
    Obtain trajectory realization $(\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$
    Compute the TD and auxiliary sequence $z_{t+1}$ [cf. (11)]:
        $\delta_t = r(\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t) + \gamma V_t(\mathbf{y}_t) - V_t(\mathbf{x}_t),$
        $z_{t+1} = (1 - \beta_t) z_t + \beta_t \delta_t$
    Compute unconstrained functional SQG step [cf. (12)]
    $\tilde{V}_{t+1}(\cdot) = (1 - \alpha_t\lambda)\tilde{V}_t(\cdot) - \alpha_t(\gamma\kappa(\mathbf{y}_t, \cdot) - \kappa(\mathbf{x}_t, \cdot)) z_{t+1}$
    Revise dict. $\tilde{\mathbf{D}}_{t+1} = [\mathbf{D}_t,\ \mathbf{x}_t\ \ , \mathbf{y}_t]$, weights $\tilde{\mathbf{w}}_{t+1} \leftarrow [(1 - \alpha_t\lambda)\mathbf{w}_t,\ \ \alpha_t z_{t+1}, -\alpha_t\gamma z_{t+1}]$
    Compress dictionary via Alg. 2, obtain coeffs. via (24)
        $(V_{t+1}, \mathbf{D}_{t+1}, \mathbf{w}_{t+1}) = \textbf{KOMP}(\tilde{V}_{t+1}, \tilde{\mathbf{D}}_{t+1}, \tilde{\mathbf{w}}_{t+1}, \epsilon_t)$
  end for
```

defined at the beginning of this subsection (15). Next we discuss how this update amounts to modifications of the parametric updates (14) defined by functional SQG. These subspace projections may be computed efficiently by exploiting the kernel parameterization described in Appendix A operating together with destructive matching pursuit [48].

We summarize the overall method, Parsimonious Kernel Gradient Temporal Difference (PKGTD) in Algorithm 1: we execute the stochastic projection of the functional SQG iterates onto sparse subspaces $\mathcal{H}_{\mathbf{D}_{t+1}}$ stated in (17). With initial function null $V_0 = 0$ (empty dictionary $\mathbf{D}_0 = []$ and coefficients $\mathbf{w}_0 = []$),at each step, given an i.i.d. sample $(\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ and step-sizes α_t, β_t, we compute the *unconstrained* functional SQG iterate $\tilde{V}_{t+1}(\cdot) = (1 - \alpha_t\lambda)\tilde{V}_t(\cdot) - \alpha_t(\gamma\kappa(\mathbf{y}_t, \cdot) - \kappa(\mathbf{x}_t, \cdot)) z_{t+1}$ parameterized by $\tilde{\mathbf{D}}_{t+1}$ and $\tilde{\mathbf{w}}_{t+1}$ as stated in (23), which are fed into KOMP (Algorithm 2 in Appendix A) with budget ϵ_t, i.e., $(V_{t+1}, \mathbf{D}_{t+1}, \mathbf{w}_{t+1}) = \text{KOMP}(\tilde{V}_{t+1}, \tilde{\mathbf{D}}_{t+1}, \tilde{\mathbf{w}}_{t+1}, \epsilon_t)$.

Remark 1. While two time-scale stochastic approximation originally appeared in the 1980s [17], [18] for compositional stochastic programming with asymptotic stability established, their role in RL, namely, to form the foundation of actor-critic algorithms [19], [38] was more recent.

A separate but related line of research in RL identifies their use not in actor-critic (which is at its core a policy search method), but instead in order to solve Bellman equations (approximate dynamic programming), beginning with [21]. In [21] the authors hypothesize that one possible reason for instability of temporal difference learning under function approximation (as detailed in [9]), is that these are *not gradient algorithms* but instead stochastic fixed point iterations. Thus, their stability interacts in a more intricate manner with the function parameterization. By reformulating Bellman equations as optimization problems, these problems are identified as possessing compositional structure, and thus are amenable to two time-scale algorithms, yielding *gradient temporal difference* learning (GTD). The derivation of PKGTD is structurally aligned with GTD in that its derivation generalizes the derivation of GTD, rather than actor-critic, although two time-scale methods are at the core of both approaches.

IV. Convergence Analysis

We now analyze the stability and memory requirements of Algorithm 1 developed in Section III. In stochastic fixed-point methods such as TD learning, the interplay between the Bellman operator contraction [5] and expectations prevents the construction of supermartingales underlying stochastic descent stability [49]. Attempts to overcome this challenge based on stochastic backward-differences require the state space to be completely explored in the limit *per step* (intractable when $|\mathcal{X}| = \infty$) [50], or stipulate that data dependent matrices be non-singular [21], respectively. Thus these methods must be analyzed using ideas from dynamical systems [51]. In contrast, we establish that Algorithm 1 belongs to the family of descent algorithms, and hence its behavior can be connected to that of supermartingales [52] – to the best of our knowledge, this is the first time supermartingales have been used in analyzing stochastic methods for MDPs. This is also true of GTD [21], although it is analyzed using ODEs [51].

Under the assumptions stated in Appendix C, it is possible to derive the fact that the auxiliary variable z_t and value function estimate V_t satisfy supermartingale-type relationships, but their behavior is intrinsically coupled. We generalize recently developed coupled supermartingale tools in [52], i.e., Lemma 2 in Appendix D, to RKHSs in order to establish the following almost sure convergence result when the step-sizes and compression budget are diminishing.

Theorem 1. *Consider the sequence z_t [cf.* (11)*] and $\{V_t\}$ [cf. 15] as stated in Algorithm 1. Assume the regularizer is positive $\lambda > 0$, Assumptions 2 - 4 hold, with the step-size conditions:*

$$\sum_{t=1}^{\infty}\alpha_t = \infty\ ,\ \sum_{t=1}^{\infty}\beta_t = \infty,\ \sum_{t=1}^{\infty}\alpha_t^2 + \beta_t^2 + \frac{\alpha_t^2}{\beta_t} < \infty\ ,\ \epsilon_t = \alpha_t^2 \quad (18)$$

Then $V_t \to V^*$ [cf. (8)*] with probability* 1*, and thus achieves the regularized Bellman fixed point* (4) *restricted to the RKHS.*

The proof is given in Appendix E. Theorem 1 states that the value functions generated by Algorithm 1 converge a.s. to the optimal V^* defined by (8). With regularizer λ made small but nonzero, using a universal kernel (e.g., a Gaussian), V_t converges close to a function satisfying Bellman's equation in *infinite MDPs* (3). By decreasing the regularizer, limiting solutions close in on those which satisfy Bellman's equation, though precise notions of closeness require continuity which is difficult to verify, given an arbitrary bounded reward. This is the first guarantee w.p.1 for a true stochastic descent method with an infinitely and nonlinearly parameterized value function. Theorem 1 requires attenuating step-sizes such that the stochastic approximation error approaches null. In contrast, constant learning rates allow for maintaining algorithm adaptivity, motivating the following result.

One step-size sequence which satisfies the attenuation conditions (18) is $\alpha_t = \mathcal{O}(t^{-(3/4+\zeta/2)})$ $,\beta_t =$

$\mathcal{O}(t^{-(1+\zeta)/2})$ $,\epsilon_t = \mathcal{O}(\alpha_t^2) = \mathcal{O}(t^{-(3/2+\zeta)})$, where $\zeta > 0$ is an arbitrarily small constant so that series $\sum_t \alpha_t$ and $\sum_t \beta_t$ diverge. Generally, satisfying (18), requires: $\alpha_t = \mathcal{O}(t^{-p_\alpha})$, $\beta_t = \mathcal{O}(t^{-p_\beta})$ with $p_\alpha \in (3/4, 1)$ and $p_\beta \in (1/2, 2p_\alpha - 1)$.

Theorem 2. *Suppose Algorithm 1 is run with constant positive learning rates $\alpha_t = \alpha$ and $\beta_t = \beta$ and constant compression budget $\epsilon_t = \epsilon$ with sufficiently large regularization, i.e.*

$$0 < \beta < 1 \ , \alpha = \beta, \epsilon = C\alpha^2, \lambda = G_V^2 + \lambda_0 \tag{19}$$

where $C > 0$ is a scalar, and $0 < \lambda_0 < 1$. Then, under Assumptions 2 - 4, the sub-optimality sequence $\|V_t - V^*\|_{\mathcal{H}}^2$ converges in mean to a neighborhood:

$$\liminf_{t\to\infty} \mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] = \mathcal{O}\left(\alpha\right) . \tag{20}$$

Theorem 2 (proof in Appendix F) establishes that the value function estimates generated by Algorithm 1 converge in expectation to a neighborhood when constant step-sizes α and β and sparsification budget ϵ in Algorithm 2 are small constants. In particular, the bias ϵ induced by sparsification does not cause instability even when it is *not going to null*. Moreover, this result only holds when the regularizer λ is chosen large enough, which numerically induces a forgetting factor on past kernel dictionary weights (23). We may make the learning rates α and β arbitrarily small, which yield a proportional decrease in the radius of convergence to a neighborhood of the Bellman fixed point (3).

In general, Theorem 2 *does not* imply the sequence actually converges, but only that its $\liminf$ converges. To establish that the entire sequence converges, even in expectation, when used with constant step-sizes, one must recursively average the functions and apply convexity of the objective function to establish the error bound decreases with the final iteration, as in Polyak-Ruppert averaging [53]. Averaging, however, will be afflicted by the fact that different functions in the RKHS do not belong to the same subspace, and therefore their kernel dictionaries will need to be pooled, causing the model order to spike. Thus, averaging is a technique of theoretical interest only in establishing limiting genuine behavior in RKHSs, and cannot be used unless parsimony is not a consideration. By contrast, under constant step-sizes selection, the value function estimates have moderate complexity in the worst case.

As noted in Section III, the complexity of functional stochastic quasi-gradient method in a RKHS is of order $\mathcal{O}(2(t-1))$ which grows without bound. To surmount this challenge, we propose subspace projections in Section III-A. We formalize here that this projection indeed controls complexity when constant learning rates and compression budget are used. This result is a corollary, as it extends Theorem 3 in [32]. To obtain this result (proof in Appendix G), we require the reward function to be bounded, as we state next.

Assumption 1. *The reward $r : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to \mathbb{R}$ is bounded:*

$$r(\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t) \leq R_{\max} \ \textit{for all} \ t, \mathbf{x}, \mathbf{a}, \mathbf{y} \tag{21}$$

Assumption 1 holds whenever the reward function is continuous and the state and action spaces are compact, and thus holds for many popular RL problems. In this setting, the complexity of Algorithm 1 is finite, as is formalized next.

Corollary 1. *Denote V_t as the value function of Algorithm 1 with constant step-sizes $\alpha_t = \alpha$ and $\beta_t = \beta \in (0, 1)$ with compression budget $\epsilon_t = \epsilon = C\alpha^2$ and regularization $\lambda = (\alpha/\beta)G_V^2 + \lambda_0 = \mathcal{O}(\alpha\beta^{-1} + 1)$ as in Remark 2. Let M_t be its associated model order, i.e., the number of columns in its dictionary. Then there exists a finite upper bound M^∞ such that, for all $t \geq 0$, the model order is bounded $M_t \leq M^\infty$.*

Under specific selections (19), the algorithm converges to a neighborhood of the optimal value function, whose radius depends on the step-sizes, and may be made small by decreasing α at the cost of a decreasing learning rate. More importantly, the use of constant step-sizes and compression budget with large enough regularization yields a value function parameterized by a dictionary whose model order is always bounded (Corollary 1). Thus, we may converge to an optimal neighborhood while ensuring the memory of the function parameterization is under control, and in the worst-case related to the covering number (metric entropy) of the state space.

V. Experiments

Our experiments aim to compare PKGTD to other policy evaluation techniques in this domain. Because it seeks memory-efficient solutions over an RKHS, we expect PKGTD to obtain accurate estimates of the value function using only a fraction of the memory required by the other methods. We perform experiments on the classical Mountain Car domain [1]: an agent applies discrete actions $\mathcal{A} = \{\texttt{reverse}, \texttt{coast}, \texttt{forward}\}$ to a car that starts at the bottom of a valley and attempts to climb up to a goal at the top of one of the mountain sides. The state space is continuous, consisting of the car's scalar position and velocity, i.e., $\mathcal{X} \subset \mathbb{R}^2$. The reward function $r(\mathbf{x}_t, \mathbf{a}_t, \mathbf{y}_t)$ is -1 unless $\mathbf{y}_t$ is the goal state at the mountain top, in which case it is 0 and the episode terminates.

Now we describe the configuration of the algorithms used for comparison. The Mountain Car environment has a two-dimensional state space (position and velocity) with bounds of $[-1.2, 0.6]$ in position, and $[-0.07, 0.07]$ in velocity. We chose not to normalize this state space to $[0, 1]$ intervals, choosing instead to handle the scale difference by using non-isotropic kernels. The ratio of the kernel variances is equal to the ratio of the lengths of their corresponding bounds, so they would be isotropic kernels if we normalized the state space.

We used a fixed non-isotropic kernel bandwidth of $\sigma_1 = 0.2, \sigma_2 = 0.0156$ in all cases. By fixing the kernel bandwidth across all algorithms, we are basically enforcing that the learned functions all belong to the same Kernel Hilbert Space.

For PKGTD, the relevant parameters are the step size, α, the rate of expectation update, β, the regularizer, λ, and the approximation error, K. For GPTD, the relevant parameters are the gaussian process noise standard deviation, σ_0, the linear independence test bound, ν, and the regularizer, λ. For the RBF grids fit using GTD, the relevant parameters are the grid spacing in the position and velocity directions, h_1 and h_2, respectively, the step size, α, and the rate of expectation update, β. Our values are summarized in Table 1. While theoretically the selection of λ is sensitive, experimentally we find it to

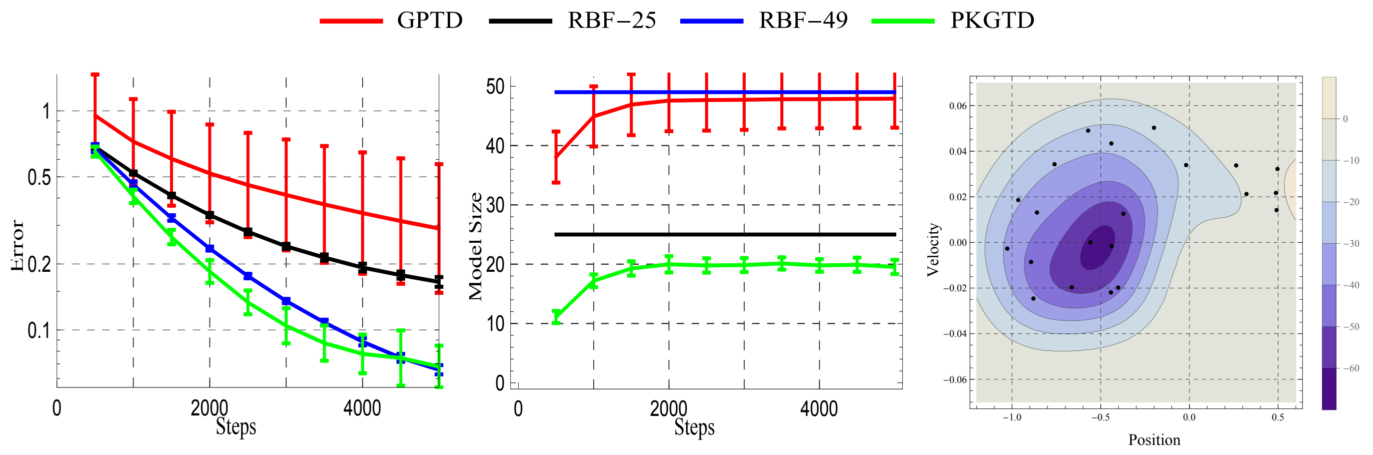

Fig. 1. Experimental comparison of PKGTD to existing kernel methods for policy evaluation on the Mountain Car task. Test set error (left), and the parameterization complexity (center) vs. iterations. PKGTD learns fastest and most stably with the least complexity (best viewed in color). We plot the contour of the learned value function (right): its minimal value is in the valley, and states near the goal are close to null. Bold black dots are kernel dictionary elements, or retained instances. Sample means and standard deviations of performance metrics (left and center) are generated via 100 individual training runs.

have little impact, and thus fix it as a small value $\lambda = 10^{-6}$.

	α	β	λ	K	σ_0	ν	h_1	h_2
PKGTD	8.0	0.2	1e-6	0.02				
GPTD			1e-6		0.01	0.2		
RBF-25	10.0	0.25					0.44	0.0343
RBF-49	1.5	0.35					0.26	0.0203

Table 1: Experimental Parameters

Theoretically and experimentally, we require that $\beta \in (0, 1)$. Selection of $\beta \approx 1/4$ was done by consulting values considered in the original GTD experiments, and $\alpha = 8$ or $\alpha = 10$ was based on the fact that larger step-sizes yield faster learning, so it is advantageous to use step-sizes α an order of magnitude larger than β.

To obtain a benchmark policy for this task, we make use of trust region policy optimization [54]. To evaluate value function estimates, we form an offline training set of state transitions and associated rewards by running this policy through consecutive episodes until we had one training trajectory of 5000 steps and then repeat this for 100 training trajectories to generate sample statistics. For ground truth, we generate one long trajectory of 10000 steps and randomly sample 2000 states from it. From each of these 2000 states, we apply the policy until episode termination and use the observed discounted return as $\hat{V}_\pi(\mathbf{x})$. Since our policy was deterministic, we only performed this procedure once per sampled state. For value function V, we define the percentage error metric: Percentage Error$(V) = (1/2000)\sum_{i=1}^{2000} |(V(\mathbf{x}_i) - \hat{V}_\pi(\mathbf{x}_i))/\hat{V}_\pi(\mathbf{x}_i)|$

We compared PKGTD with a Gaussian kernel to two other techniques for policy evaluation that also use kernel-based value function representations: (1) Gaussian process temporal difference (GPTD) [31], and (2) gradient temporal difference (GTD) [21] using radial basis function (RBF) network features. We fix a kernel bandwidth across all techniques, and select parameter values that yield the best results for each method (see Table 1). For RBF feature generation, we use two fixed grids with different spacing. The first was one for which GTD yielded a value function estimate with percentage error similar to that which we obtained using PKGTD (RBF-49), and the second was one which yielded a number of basis functions that was similar to what PKGTD selected (RBF-25).

Figure 1 displays these results: on the left we show percentage error, a surrogate for Bellman evaluation error, versus training example, in which we observe that PKGTD yields fast and reliable learning. In the center figure, we show the number of points in the kernel dictionary (model size) over samples, which demonstrates that PKGTD only keeps past states needed to estimate the value function well, rather than statistically insignificant points. Overall, we note that GTD with fixed RBF features requires a much denser grid in order to reach the same Percentage Error as Algorithm 1, and that adaptive instance selection results in both faster initial learning and smaller error. Compared to GPTD, which chooses model points online according to a fixed linear-dependence criterion, PKGTD requires fewer model points and converges to a better estimate of the value function more quickly and stably.

Fig. 1 (right) displays a contour plot of the value function – the x-axis denotes position, the y-axis denotes velocity, and bold black dots denote kernel dictionary elements, i.e., past visited states that are essential for representing the estimate of V^π. The contour plot suggests that low value states are when one has small velocity near position -0.6 (the bottom of the hill). Moreover, value progressively increases as speed increases away from the bottom of the hill towards the top.

VI. Conclusion

In this paper, we considered the problem of policy evaluation in infinite MDPs with value functions that belong to a RKHS. To solve this problem, we extended recent SQG methods for compositional stochastic programming to a RKHS, and used the result, combined with greedy sparse subspace projection, in a new policy-evaluation procedure called PKGTD (Algorithm 1). Under diminishing step sizes, PKGTD solves Bellman's evaluation equation exactly under the hypothesis that its fixed point belongs to a RKHS (Theorem 1). Under constant step sizes, we can further guarantee finite-memory approximations (Corollary 1) that still exhibit mean convergence to a neighborhood of the optimal value function (Theorem 2). In our Mountain Car experiments, PKGTD yields excellent sample efficiency and model complexity, and therefore holds promise for large state space problems common in robotics where fixed state-action space tiling may prove impractical.

References

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

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robotics Res.*, p. 0278364913495721, 2013.

[3] W. R. Scott, W. B. Powell, and S. Moazehi, "Least squares policy iteration with instrumental variables vs. direct policy search: Comparison against optimal benchmarks using energy storage," *arXiv preprint arXiv:1401.0843*, 2014.

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

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

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

[7] S. J. Bradtke and A. G. Barto, "Linear least-squares algorithms for td learning," *Machine learning*, vol. 22, no. 1-3, pp. 33–57, 1996.

[8] 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," *J. of Control Theory & Apps.*, vol. 9, no. 3, pp. 336–352, 2011.

[9] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Autom. Control*, vol. 42, no. 5, pp. 674–690, 1997.

[10] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *ICML*. ACM, 2008, pp. 664–671.

[11] 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 *NeurIPS*, 2009, pp. 1204–1212.

[12] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *ICML*. Morgan Kaufmann, 1995, pp. 30–37.

[13] N. K. Jong and P. Stone, "Model-based function approximation in reinforcement learning," in *AAMAS*. ACM, 2007, p. 95.

[14] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *Departmental Papers (ESE)*, p. 323, 1992.

[15] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *ICML*, 1999, pp. 278–287.

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

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

[18] Y. Ermoliev, "Stochastic quasigradient methods and their application to system optimization," *Stochastics*, vol. 9, no. 1-2, pp. 1–36, 1983.

[19] V. S. Borkar and V. R. Konda, "The actor-critic algorithm as multi-time-scale stochastic approximation," *Sadhana*, vol. 22, no. 4, pp. 525–543, 1997.

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

[21] 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 *NeurIPS*, 2009, pp. 1609–1616.

[22] G. Kimeldorf and G. Wahba, "Some results on tchebycheffian spline functions," *J Math Anal Appl*, vol. 33, no. 1, pp. 82–95, 1971.

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

[24] G. Taylor and R. Parr, "Kernelized value function approximation for reinforcement learning," in *ICML*. ACM, 2009, pp. 1017–1024.

[25] S. Grünewälder, G. Lever, L. Baldassarre, M. Pontil, and A. Gretton, "Modelling transition dynamics in mdps with rkhs embeddings," in *ICML*, vol. 1, 2012, pp. 535–542.

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

[27] B. Dai, N. He, Y. Pan, B. Boots, and L. Song, "Learning from conditional distributions via dual embeddings," in *Artificial Intelligence and Statistics*, 2017, pp. 1458–1467.

[28] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition," in *Asilomar Conference*, 1993.

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

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

[31] Y. Engel, S. Mannor, and R. Meir, "Bayes meets bellman: The gaussian process approach to temporal difference learning," in *ICML*, 2003.

[32] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," *JMLR*, vol. 20, no. 1, pp. 83–126, 2019.

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

[34] M. Wang, J. Liu, and E. Fang, "Accelerating stochastic composition optimization," in *NeurIPS*, 2016, pp. 1714–1722.

[35] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *JMLR*, vol. 7, no. Dec, pp. 2651–2667, 2006.

[36] E. Tolstaya, A. Koppel, E. Stump, and A. Ribeiro, "Nonparametric stochastic compositional gradient descent for q-learning in continuous markov decision problems," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 6608–6615.

[37] D. Precup, "Eligibility traces for off-policy policy evaluation," *Computer Science Department Faculty Publication Series*, p. 80, 2000.

[38] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control & Optimization*, vol. 42, no. 4, pp. 1143–1166, 2003.

[39] H. Kumar, A. Koppel, and A. Ribeiro, "On the sample complexity of actor-critic method for reinforcement learning with function approximation," *arXiv preprint arXiv:1910.08412*, 2019.

[40] K. Zhang, A. Koppel, H. Zhu, and T. Başar, "Global convergence of policy gradient methods to (almost) locally optimal policies," *arXiv preprint arXiv:1906.08383*, 2019.

[41] S. Bhatt, A. Koppel, and V. Krishnamurthy, "Policy gradient using weak derivatives for reinforcement learning," in *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, March 2019, pp. 1–3.

[42] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *ICML*. ACM, 2009, pp. 993–1000.

[43] 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.

[44] R. Wheeden, R. Wheeden, and A. Zygmund, *Measure and Integral: An Introduction to Real Analysis*, ser. Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis, 1977. [Online]. Available: https://books.google.com/books?id=YDkDmQ_hdmcC

[45] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online Learning with Kernels," *IEEE Trans. Signal Process.*, vol. 52, pp. 2165–2176, August 2004.

[46] A. Smola, A. Gretton, L. Song, and B. Schölkopf, "A hilbert space embedding for distributions," in *International Conference on Algorithmic Learning Theory*. Springer, 2007, pp. 13–31.

[47] D.-X. Zhou, "The covering number in learning theory," *Complexity*, vol. 18, no. 3, pp. 739–767, 2002.

[48] P. Vincent and Y. Bengio, "Kernel matching pursuit," *Machine Learning*, vol. 48, no. 1, pp. 165–187, 2002.

[49] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951.

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

[51] V. S. Borkar and S. P. Meyn, "The ode method for convergence of stochastic approximation and reinforcement learning," *SICON*, vol. 38, no. 2, pp. 447–469, 2000.

[52] M. Wang and D. P. Bertsekas, "Incremental constraint projection-proximal methods for nonsmooth convex optimization," *SIOPT*, 2014.

[53] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SICON*, vol. 30, no. 4, pp. 838–855, 1992.

[54] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *ICML*, 2015, pp. 1889–1897.

[55] D. Needell, J. Tropp, and R. Vershynin, "Greedy signal recovery review," in *Asilomar Conference*. IEEE, 2008, pp. 1048–1050.

[56] H. Brezis, *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[57] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug 2004.

Supplementary for IEEE-TAC 19-2021: Policy Evaluation in Continuous MDPs with Efficient Kernelized Gradient Temporal Difference

by Alec Koppel, Garrett Warnell, Ethan Stump, Alejandro Ribeiro, and Peter Stone

Algorithm 2 Destructive Kernel Orthogonal Matching Pursuit (KOMP)

```
Require: function $\tilde{V}$ defined by dict. $\tilde{\mathbf{D}} \in \mathbb{R}^{p\times\tilde{M}}$, coeffs. $\tilde{\mathbf{w}} \in \mathbb{R}^{\tilde{M}}$, approx. budget $\epsilon_t > 0$
  initialize $V = \tilde{V}$, dict. $\mathbf{D} = \tilde{\mathbf{D}}$ with indices $\mathcal{I}$, model order $M = \tilde{M}$, coeffs. $\mathbf{w} = \tilde{\mathbf{w}}$.
  while candidate dictionary is non-empty $\mathcal{I} \neq \emptyset$ do
    for $j = 1, \dots, \tilde{M}$ do
      Find minimal approx. error without dict. element $\mathbf{d}_j$
        $\gamma_j = \min_{\mathbf{w}_{\mathcal{I}\setminus\{j\}}\in\mathbb{R}^{M-1}} \|\tilde{V}(\cdot) - \sum_{k\in\mathcal{I}\setminus\{j\}} w_k\kappa(\mathbf{d}_k,\cdot)\|_{\mathcal{H}}$ .
    end for
    Find index minimizing error: $j^* = \text{argmin}_{j\in\mathcal{I}}\gamma_j$
     if  minimal error exceeds threshold $\gamma_{j^*} > \epsilon_t$
       stop
     else
       Prune dictionary $\mathbf{D} \leftarrow \mathbf{D}_{\mathcal{I}\setminus\{j^*\}}$
       Revise set $\mathcal{I} \leftarrow \mathcal{I}\setminus\{j^*\}$, model order $M \leftarrow M-1$.
       Compute updated weights $\mathbf{w}$ defined by dict. $\mathbf{D}$
         $\mathbf{w} = \underset{\mathbf{w}\in\mathbb{R}^M}{\text{argmin}}\|\tilde{V}(\cdot) - \mathbf{w}^T\boldsymbol{\kappa}_{\mathbf{D}}(\cdot)\|_{\mathcal{H}}$
     end
  end while
  return  $V, \mathbf{D}, \mathbf{w}$ of model order $M \leq \tilde{M}$ s.t. $\|V - \tilde{V}\|_{\mathcal{H}} \leq \epsilon_t$
```

Appendix

A. Kernel Parameterization

Coefficient update The update (15), for a fixed dictionary $\mathbf{D}_{t+1} \in \mathbb{R}^{p\times M_{t+1}}$, may be expressed in terms of the parameter space of coefficients only. To do so, first define the stochastic quasi-gradient update *without projection*, given function V_t parameterized by dictionary $\mathbf{D}_t$ and coefficients $\mathbf{w}_t$, as

$$\tilde{V}_{t+1} = (1-\alpha_t\lambda)V_t - \alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot) - \kappa(\mathbf{x}_t,\cdot))z_{t+1} \ . \quad (22)$$

This update may be represented using dictionary and weights

$$\begin{aligned}\tilde{\mathbf{D}}_{t+1} &= [\mathbf{D}_t \ , \ \mathbf{x}_t \ , \ \mathbf{y}_t] \\ \tilde{\mathbf{w}}_{t+1} &= [(1-\alpha_t\lambda)\mathbf{w}_t \ , \ \alpha_t z_{t+1} \ , \ -\alpha_t\gamma z_{t+1}] \ , \end{aligned} \quad (23)$$

Here we drop the time index for notational clarity but note that it can be easily inferred from the context. V_{t+1} denotes the projected SQG iterates [cf. (15)] and whereas $\tilde{V}_{t+1}$ denotes the un-projected iterate [cf. (22)] in Sec. III-A. The later is parameterized by dictionary $\tilde{\mathbf{D}}_{t+1}$ and weights $\tilde{\mathbf{w}}_{t+1}$ (23).

When the dictionary defining V_{t+1} is assumed fixed, we use the Representer Theorem to rewrite (17) as a kernel expansions, where the coefficients are the only free parameter:

$$\underset{\mathbf{w}\in\mathbb{R}^{M_{t+1}}}{\text{argmin}} \frac{1}{2\eta_t}\Big\|\sum_{n=1}^{M_{t+1}} w_n\kappa(\mathbf{d}_n,\cdot) - \sum_{m=1}^{\tilde{M}}\tilde{w}_m\kappa(\tilde{\mathbf{d}}_m,\cdot)\Big\|^2_{\mathcal{H}} \quad (24)$$
$$:= \mathbf{w}_{t+1} \ .$$

In (24), the first equality comes from expanding the square, and the second comes from defining the cross-kernel matrix $\mathbf{K}_{\mathbf{D}_{t+1},\tilde{\mathbf{D}}_{t+1}}$ whose $(n,m)^{\text{th}}$ entry is $\kappa(\mathbf{d}_n,\tilde{\mathbf{d}}_m)$. Kernel matrices $\mathbf{K}_{\tilde{\mathbf{D}}_{t+1},\tilde{\mathbf{D}}_{t+1}}$ and $\mathbf{K}_{\mathbf{D}_{t+1},\mathbf{D}_{t+1}}$ are similarly defined. Here M_{t+1} is the number of columns in $\mathbf{D}_{t+1}$, while $\tilde{M}_{t+1} = M_t + 2$ is that of in $\tilde{\mathbf{D}}_{t+1}$ [cf. (23)]. Observe that $\tilde{\mathbf{D}}_{t+1}$ has $\tilde{M}_{t+1} = M_t + 2$ columns, which is the length of $\tilde{\mathbf{w}}_{t+1}$. For a fixed dictionary $\mathbf{D}_{t+1}$, the stochastic projection in (17) is a least-squares problem on the coefficient vector, i.e.,

$$\mathbf{w}_{t+1} = \mathbf{K}^{-1}_{\mathbf{D}_{t+1}\mathbf{D}_{t+1}}\mathbf{K}_{\mathbf{D}_{t+1}\tilde{\mathbf{D}}_{t+1}}\tilde{\mathbf{w}}_{t+1} \ , \quad (25)$$

The explicit solution of (24) may be obtained by noting that the last factor is independent of $\mathbf{w}$, and thus by computing gradients and solving for $\mathbf{w}_{t+1}$ we obtain (25). Now we turn to dictionary selection $\mathbf{D}_{t+1}$ from trajectory $\{\mathbf{x}_u, \pi(\mathbf{x}_u), \mathbf{y}_u\}_{u\leq t}$.

Dictionary Update We select dictionary $\mathbf{D}_{t+1}$ via greedy compression, a topic studied in compressive sensing [55]. The function $\tilde{V}_{t+1} = (1-\alpha_t)V_t - \alpha_t(\gamma\kappa(\mathbf{y}_t,\cdot) - \kappa(\mathbf{x}_t,\cdot))z_{t+1}$ defined by SQG method without projection (22) is parameterized by dictionary $\tilde{\mathbf{D}}_{t+1}$ [cf. (23)]. We form $\mathbf{D}_{t+1}$ by selecting a subset of M_{t+1} columns from $\tilde{\mathbf{D}}_{t+1}$ that best approximate $\tilde{V}_{t+1}$ in terms of Hilbert norm error. This specification may be met via *kernel orthogonal matching pursuit* (KOMP) [48] with error tolerance ϵ_t, which yields a dictionary $\mathbf{D}_{t+1}$ comprised of a subset of columns of $\tilde{\mathbf{D}}_{t+1}$. We tune ϵ_t to ensure both descent (Lemma 1(ii)) and finite memory (Corollary 1).

With respect to the KOMP procedure above, we specifically use a variant called destructive KOMP with pre-fitting (see [48], Section 2.3). This flavor of KOMP takes as an input a candidate function $\tilde{V}$ of model order $\tilde{M}$ parameterized by its dictionary $\tilde{\mathbf{D}} \in \mathbb{R}^{p\times\tilde{M}}$ and coefficients $\tilde{\mathbf{w}} \in \mathbb{R}^{\tilde{M}}$. The method then approximates $\tilde{V}$ by $V \in \mathcal{H}$ with a lower model order. Initially, the candidate is the original $V = \tilde{V}$ so that its dictionary is initialized with $\mathbf{D} = \tilde{\mathbf{D}}$, with coefficients $\mathbf{w} = \tilde{\mathbf{w}}$. Then, we sequentially and greedily remove model points from initial dictionary $\tilde{\mathbf{D}}$ until threshold $\|V - \tilde{V}\|_{\mathcal{H}} \leq \epsilon_t$ is violated. The result is a sparse approximation V of $\tilde{V}$. Moreover, we also assume that the V_{t+1} output from KOMP has bounded Hilbert norm, which is often required in the analysis of stochastic optimization algorithms. This assumption can be explicitly enforced by adding a bounded norm constraint into the the optimization problem for finding the best set of bases in the matching pursuit algorithm, which attainable by thresholding the coefficient sequence during compression.

This process is executed via destructive KOMP. At each stage, a single dictionary element j of $\mathbf{D}$ is selected to be removed which contributes the least to the Hilbert-norm approximation error $\min_{V\in\mathcal{H}_{\mathbf{D}\setminus\{j\}}}\|\tilde{V} - V\|_{\mathcal{H}}$ of the original function $\tilde{V}$, when dictionary $\mathbf{D}$ is used. Since at each stage the kernel dictionary is fixed, this amounts to a computation involving weights $\mathbf{w} \in \mathbb{R}^{M-1}$ only; that is, the error of removing dictionary point $\mathbf{d}_j$ is computed for each j as $\gamma_j = \min_{\mathbf{w}_{\mathcal{I}\setminus\{j\}}\in\mathbb{R}^{M-1}} \|\tilde{V}(\cdot) - \sum_{k\in\mathcal{I}\setminus\{j\}} w_k\kappa(\mathbf{d}_k,\cdot)\|$. $\mathbf{w}_{\mathcal{I}\setminus\{j\}}$ denotes the entries of $\mathbf{w}\in\mathbb{R}^M$ restricted to the sub-vector associated with indices $\mathcal{I}\setminus\{j\}$. Then, we define the dictionary element which contributes the least to the approximation error as $j^* = \operatorname{argmin}_j \gamma_j$. If the error associated with removing this kernel dictionary element exceeds the given approximation budget $\gamma_{j^*} > \epsilon_t$, the algorithm terminates. Otherwise, this dictionary element $\mathbf{d}_{j^*}$ is removed, the weights $\mathbf{w}$ are revised based on the pruned dictionary as $\mathbf{w} = \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^M}\|\tilde{f}(\cdot) - \mathbf{w}^T\boldsymbol{\kappa}_{\mathbf{D}}(\cdot)\|_{\mathcal{H}}$, and the process repeats as long as the current function approximation is defined by a nonempty dictionary. See Algorithm 2 for a summary.

B. Extended Discussion

Remark 2. (Aggressive Constant Learning Rates) In practice, one may obtain better performance by using larger constant step-sizes. To do so, the criterion (19) may be relaxed: we require $0<\beta<1$ but $\alpha>0$ may be any positive scalar. Then, with regularizer chosen as $\lambda = G_V^2\frac{\alpha}{\beta} + \lambda_0$ for $0<\lambda_0<1$, the radius of convergence is (see Appendix F)

$$\liminf_{t\to\infty}\mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] = \mathcal{O}\Big(\alpha^2+\beta^2+\frac{\alpha^2}{\beta}\Big[1+\alpha^2+\frac{\alpha}{\beta}+\frac{\alpha^2}{\beta^2}\Big]\Big)\;. \qquad (26)$$

The ratios α^2/β and α^2/β^2 dominate (26) and must be made small to obtain accurate solutions.

Remark 3. (Regularization Path) In Theorem 1, we establish convergence for any $\lambda>0$ when step-sizes attenuate. That regularizer λ may not be null means that we do not extract the exact Bellman fixed point restricted to the RKHS, but only a function that is close. In related work the minimizer V_λ^* continuously depends on λ. It's beyond the scope of this work to extend these results to this setting, but on the hypothesis that they generalize to (8), we may claim that decreasing λ causes V_λ^* to be closer to fixed point stated in (3).

On the other hand, for Theorem 2 with given regularizer λ, imposes explicit restrictions on the choice of constant algorithm step-sizes. That is, we require $\lambda = G_V^2\frac{\alpha}{\beta}+\lambda_0$ for $0<\lambda_0<1$, where $\alpha>0$ and $0<\beta<1$. It is possible to derive the fact that larger regularization means faster learning rates but to less accurate solutions, in either diminishing or constant step-size settings, but these facts, which depend on rate analyses, are left to future work.

C. Technical Assumptions and Definitions

Before continuing, we introduce a few key assumptions and definitions which are required to establish convergence. In particular, for further reference, we define the functional stochastic quasi-gradient of the regularized objective as

$$\begin{aligned}\hat{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t,\pi(\mathbf{x}_t),\mathbf{y}_t) &=\\ (\gamma\kappa(\mathbf{y}_t,\cdot) &- \kappa(\mathbf{x}_t,\cdot))z_{t+1} + \lambda V_t\;, \end{aligned}\qquad (27)$$

and its sparse-subspace projected variant as

$$\begin{aligned}&\tilde{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t,\pi(\mathbf{x}_t),\mathbf{y}_t) \qquad (28)\\ &=\frac{1}{\alpha_t}\Big(V_t - \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}\Big[V_t - \alpha_t\hat{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t,\pi(\mathbf{x}_t),\mathbf{y}_t)\Big]\Big)\;,\end{aligned}$$

Note that the update (15), using (28), may be rewritten as a stochastic projected quasi-gradient step rather than a stochastic quasi-gradient step followed by set projection, i.e.,

$$V_{t+1} = V_t - \alpha_t\tilde{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t,\pi(\mathbf{x}_t),\mathbf{y}_t)\;, \qquad (29)$$

Further, define the time-dependent sigma algebra, i.e., filtration, as $\mathcal{F}_t \supset (\{V_s, z_s, \mathbf{x}_s, \pi(\mathbf{x}_s), \mathbf{y}_s\}_{s=0}^{t-1})$. Now we are ready to state the technical conditions required for convergence. All statements involving conditional expectations are imposed with probability 1, unless otherwise stated.

Assumption 2. *The state space $\mathcal{X}\subset\mathbb{R}^p$ and action space $\mathcal{A}\subset\mathbb{R}^q$ are compact, and the reproducing kernel map may be bounded as*

$$\sup_{\mathbf{x}\in\mathcal{X}}\sqrt{\kappa(\mathbf{x},\mathbf{x})} = X < \infty \qquad (30)$$

Assumption 3. *The temporal difference δ and auxiliary sequence z [cf.* (11)*] satisfy the zero-mean and finite conditional variance conditions, respectively,*

$$\begin{aligned}&\mathbb{E}\left[\delta\,\middle|\,\mathbf{x},\pi(\mathbf{x})\right] = \bar{\delta}\;, \qquad \mathbb{E}\left[(\delta-\bar{\delta})^2\,\middle|\,\mathcal{F}_t\right]\le\sigma_\delta^2\;,\\ &\mathbb{E}\left[z^2\,\middle|\,\mathbf{x},\pi(\mathbf{x})\right]\le G_\delta^2\;. \end{aligned}\qquad (31)$$

where σ_δ and G_δ are positive scalars.

Assumption 4. *The stochastic quasi-gradient, when evaluated at $\bar{\delta}$, is an unbiased estimate for the true gradient $\nabla_V J(V)$. Moreover, the difference of reproducing kernels expression (the first factor in* (10)*) has finite conditional variance:*

$$\begin{aligned}&\mathbb{E}\left[(\gamma\kappa(\mathbf{y},\cdot)-\kappa(\mathbf{x},\cdot))\bar{\delta}\right] = \nabla_V J(V)\;,\\ &\mathbb{E}\left[\|\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot)\|_{\mathcal{H}}^2\,\middle|\,\mathcal{F}_t\right]\le G_V^2\;. \end{aligned}\qquad (32)$$

Additionally, the projected stochastic gradient of the objective [cf. (28)*] has finite second conditional moment as*

$$\mathbb{E}\left[\|\tilde{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t,\pi(\mathbf{x}_t),\mathbf{y}_t)\|_{\mathcal{H}}^2\,\middle|\,\mathcal{F}_t\right]\le\sigma_V^2\;, \qquad (33)$$

and the conditional mean of the temporal difference $\bar{\delta}$ is Lipschitz continuous with respect to the value function V, i.e for any two distinct δ and $\tilde{\delta}$, we have

$$|\bar{\delta}-\bar{\tilde{\delta}}|\le L_V\|V-\tilde{V}\|_{\mathcal{H}} \qquad (34)$$

where $V,\tilde{V}\in\mathcal{H}$ are distinct RKHS elements, $L_V>0$ is a scalar, and $\bar{\delta} = \mathbb{E}_{\mathbf{y}}[r(\mathbf{x},\pi(\mathbf{x}),\mathbf{y})+\gamma V(\mathbf{y}) - V(\mathbf{x})\,|\,\mathbf{x},\pi(\mathbf{x})]$.

Assumption 2 regarding the compactness of the state and action spaces of the Markov Decision Process intrinsically hold for most application settings and limit the radius of the set from which the MDP trajectory is sampled. Similar boundedness conditions on the reproducing kernel map have been considered in supervised learning applications [45]. The mean and variance properties of the temporal difference stated in Assumption 3 to bound the error in the descent direction associated with stochastic approximations, and are necessary to establish stability of stochastic methods. Assumption 4 is similar to Assumption 3 but instead of establishing bounds on the stochastic approximation error of the temporal difference, limits stochastic error variance in the reproducing kernel Hilbert space. These are natural extensions of the conditions needed for convergence of stochastic compositional gradient methods with vector-valued decision variables [33]. However, we note that (34), in the context of MDPs, restricts the class of reward functions to be those which may be smoothly interpolated in a RKHS. This condition holds, for instance, when the reward is a potential or navigation-like function [14], [15], which are well-known to interact favorably in the design of controllers from a dynamical systems perspective.

The stipulation that the KOMP projection explicitly thresholds the norm of the value functions allows us to write

$$\|V_t\|_{\mathcal{H}} \leq K \;, \qquad \|V^*\|_{\mathcal{H}} \leq K \;, \qquad \text{for all } t \tag{35}$$

where $K > 0$ is some constant. The boundedness of V^* follows from the fact that since $\mathcal{X}$ is compact and $J(V)$ is a continuous convex function over a compact set, its minimizer is achieved over this compact set [56][Corrolary 3.23].

D. Auxiliary Results and Technical Lemmas

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

Proposition 1. *Given independent identical realizations* $(\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ *of the random triple* $(\mathbf{x}, \pi(\mathbf{x}), \mathbf{y})$*, the difference between the projected stochastic functional quasi-gradient and the stochastic functional quasi-gradient of the instantaneous cost instantaneous risk defined by* (27) *and* (28)*, respectively, is bounded for all* t *as*

$$\|\tilde{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t, \pi(\mathbf{x}_t),\mathbf{y}_t) - \hat{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t, \pi(\mathbf{x}_t),\mathbf{y}_t)\|_{\mathcal{H}} \leq \frac{\epsilon_t}{\alpha_t} \tag{36}$$

where $\alpha_t > 0$ *denotes the algorithm step-size and* $\epsilon_t > 0$ *is the compression budget parameter of Algorithm 2.*

Proof: As in Proposition 6 of [32], consider the square-Hilbert-norm difference of $\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ [cf. (27)] and $\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ [cf. (28)]

$$\begin{aligned}
&\|\tilde{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t, \pi(\mathbf{x}_t),\mathbf{y}_t) - \hat{\nabla}_V J(V_t, z_{t+1};\mathbf{x}_t, \pi(\mathbf{x}_t),\mathbf{y}_t)\|_{\mathcal{H}} \\
&\quad = \Big\| \frac{1}{\alpha}\Big(V_t - \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}\Big[V_t - \alpha_t \hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\Big]\Big) \\
&\qquad - \hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\Big\|_{\mathcal{H}}^2
\end{aligned} \tag{37}$$

Multiply and divide $\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$, the last term, by α_t, and reorder terms to write

$$\begin{aligned}
&\Bigg\| \frac{\Big(V_t - \alpha_t \hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\Big)}{\alpha_t} \\
&\qquad - \frac{\mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}\Big[V_t - \alpha_t \hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\Big]\Big)}{\alpha_t}\Bigg\|_{\mathcal{H}}^2 \\
&= \frac{1}{\alpha_t^2}\Big\|\big(V_t - \alpha_t \hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t) \\
&\qquad - \mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}\Big[V_t - \alpha_t \hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\Big]\Big)\Big\|_{\mathcal{H}}^2 \\
&= \frac{1}{\alpha_t^2}\|\tilde{V}_{t+1} - V_{t+1}\|_{\mathcal{H}}^2 \leq \frac{\epsilon_t^2}{\alpha_t^2}
\end{aligned} \tag{38}$$

where we have pulled the nonnegative scalar α_t outside the norm on the second line and substituted the definition of $\tilde{V}_{t+1}$ and V_{t+1} in (12) and (15), respectively, in the last one. These facts combined with the KOMP residual stopping criterion in Algorithm 2 is $\|\tilde{V}_{t+1} - V_{t+1}\|_{\mathcal{H}} \leq \epsilon_t$ applied to the last term on the right-hand side of (38) yields (36).

■

Lemma 1. *Let Assumptions 2 - 4 hold true and consider the sequence of iterates defined by Algorithm 1. Then:*

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

$$\mathbb{E}\big[\|V_{t+1} - V_t\|_{\mathcal{H}}^2 \,\big|\, \mathcal{F}_t\big] \leq 2\alpha_t^2(G_\delta^2 G_V^2 + \lambda^2 K^2) + 2\epsilon_t^2 \tag{39}$$

ii) *The conditional expectation of the Hilbert-norm difference of value functions at the next and current iteration satisfies the relationship*

$$\begin{aligned}
\mathbb{E}\big[\|V_{t+1} - V^*\|_{\mathcal{H}}^2 \big| \mathcal{F}_t\big] \leq{}& \Big(1 + \frac{\alpha_t^2}{\beta_t} G_V^2\Big)\|V_t - V^*\|_{\mathcal{H}}^2 \\
&+ 2\epsilon_t \|V_t - V^*\|_{\mathcal{H}} - 2\alpha_t[J(V_t) - J(V^*)] \\
&+ \alpha_t^2 \sigma_V^2 + \beta_t \mathbb{E}\left[(z_{t+1} - \bar{\delta}_t)^2 \middle| \mathcal{F}_t\right] .
\end{aligned} \tag{40}$$

iii) *Define the expected value of the temporal difference given the state variable* $\mathbf{x}$ *and policy* π *as* $\bar{\delta}_t = \mathbb{E}[\delta_t \,|\, \mathbf{x}_t, \pi(\mathbf{x}_t)]$*. Then the evolution of the auxiliary sequence* z_t *with respect to* $\bar{\delta}_t$ *satisfies*

$$\begin{aligned}
\mathbb{E}\big[(z_{t+1} - \bar{\delta}_t)^2 \big| \mathcal{F}_t\big] \leq{}& (1 - \beta_t)(z_t - \bar{\delta}_{t-1})^2 + \frac{L_V}{\beta_t}\|V_t - V_{t-1}\|_{\mathcal{H}}^2 \\
&+ 2\beta_t^2 \sigma_\delta^2
\end{aligned} \tag{41}$$

Proof of Lemma 1(i): Consider the Hilbert-norm difference of value functions at the next and current iteration, and use the definition of V_{t+1} in (29), i.e.,

$$\begin{aligned}
\|V_{t+1} - V_t\|_{\mathcal{H}}^2 &= \alpha_t^2 \|\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|_{\mathcal{H}}^2 \\
&\leq 2\alpha_t^2 \|\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|_{\mathcal{H}}^2 \\
&\quad + 2\alpha_t^2 \|\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t) \\
&\quad - \tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|_{\mathcal{H}}^2 \;,
\end{aligned} \tag{42}$$

where we add and subtract the functional stochastic quasi-gradient $\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ on the first line of (42) and the fact that the square of a sum is less than the sum of squares, due to Cauchy-Schwartz, i.e., $(a+b)^2 \leq 2a^2+2b^2$ for any $a, b > 0$. Now, we may apply Proposition 1 to the second term and compute the conditional expectation to obtain

$$\begin{aligned}&\mathbb{E}[\|V_{t+1} - V_t\|^2_{\mathcal{H}} \,|\, \mathcal{F}_t] &(43)\\ &\quad= 2\alpha_t^2\mathbb{E}[\|\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|^2_{\mathcal{H}} \,|\, \mathcal{F}_t] + 2\epsilon_t^2 \; . &(44)\end{aligned}$$

Use the Cauchy-Schwartz inequality together with Law of Total Expectation and the definition of the functional stochastic quasi-gradient (27) to upper-estimate (43) as

$$\begin{aligned}&\mathbb{E}[\|V_{t+1}-V_t\|^2_{\mathcal{H}} \,|\, \mathcal{F}_t]\\ &\quad\leq 2\alpha_t^2\mathbb{E}\Big\{\|\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))\|^2_{\mathcal{H}}\\ &\qquad\times \mathbb{E}[z_{t+1}^2 \,|\, \mathbf{x}_t, \pi(\mathbf{x}_t)] \,|\, \mathcal{F}_t\Big\}+2\alpha_t^2\lambda\|V_t\|^2_{\mathcal{H}} + 2\epsilon_t^2 \; , \qquad (45)\end{aligned}$$

which together with equation 31 (Assumption 3) regarding fact that z_{t+1} has a finite second conditional moment, yields

$$\begin{aligned}\mathbb{E}[\|V_{t+1}-V_t\|^2_{\mathcal{H}}|\mathcal{F}_t] &\leq 2\alpha_t^2G_\delta^2\mathbb{E}\Big[\|\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot))\|^2_{\mathcal{H}}|\mathcal{F}_t\Big]\\ &\quad + 2\alpha_t^2\lambda\|V_t\| + 2\epsilon_t^2\\ &\leq 2\alpha_t^2(G_\delta^2G_V^2 + \lambda^2K^2)+2\epsilon_t^2 \; , \qquad (46)\end{aligned}$$

where we have also applied the fact that the functional gradient of the temporal difference $\gamma\kappa(\mathbf{y}_t, \cdot) - \kappa(\mathbf{x}_t, \cdot))$ has a finite second conditional moment and the bound on the function sequence [cf. (35)], allowing us to conclude (39). ■

Proof of Lemma 1(ii): This proof is a generalization of Lemma 3 in Appendix G.2 in the Supplementary Material of [33] to a function-valued stochastic quasi-gradient step combined with bias induced by the sparse subspace projections $\mathcal{P}_{\mathcal{H}_{\mathbf{D}_{t+1}}}[\cdot]$ in (15). Begin by considering the square-Hilbert norm sub-optimality of V_{t+1}, i.e.,

$$\begin{aligned}&\|V_{t+1} - V^*\|^2_{\mathcal{H}}\\ &\quad= \|V_t - \alpha_t\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t) - V^*\|^2_{\mathcal{H}}\\ &\quad= \|V_t-V^*\|^2_{\mathcal{H}}-2\alpha_t\langle\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t),V_t-V^*\rangle_{\mathcal{H}}\\ &\qquad+ \alpha_t^2\|\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|^2_{\mathcal{H}} \; , \qquad (47)\end{aligned}$$

where we use the reformulation of the projected functional stochastic quasi-gradient step defined in (29) for the first equality, and expand the square in the second. Now, adding and subtracting $\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ the (un-projected) functional stochastic quasi-gradient (27) yields

$$\begin{aligned}&\|V_{t+1} - V^*\|^2_{\mathcal{H}}\\ &\quad=\|V_t-V^*\|^2_{\mathcal{H}}-2\alpha_t\langle\hat{\nabla}_V J(V_t,z_{t+1}; \mathbf{x}_t,\pi(\mathbf{x}_t), \mathbf{y}_t),V_t-V^*\rangle_{\mathcal{H}}\\ &\qquad+ 2\alpha_t\langle\hat{\nabla}_V J(V_t,z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\\ &\qquad- \tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t), V_t-V^*\rangle_{\mathcal{H}}\\ &\qquad+ \alpha_t^2\|\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|^2_{\mathcal{H}} \; . \qquad (48)\end{aligned}$$

Apply the Cauchy-Schwartz inequality to the third term on the right-hand side of (48) together with the bound on the difference between unprojected and projected stochastic quasi-gradients in Proposition 1 to obtain

$$\begin{aligned}&\|V_{t+1} - V^*\|^2_{\mathcal{H}} \qquad (49)\\ &\quad= \|V_t-V^*\|^2_{\mathcal{H}}-2\alpha_t\langle\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t),V_t-V^*\rangle_{\mathcal{H}}\\ &\qquad+ 2\epsilon_t\|V_t - V^*\|_{\mathcal{H}} + \alpha_t^2\|\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|^2_{\mathcal{H}} \; .\end{aligned}$$

Now, with $\bar{\delta}_t = \mathbb{E}[\delta_t \,|\, \mathbf{x}_t, \pi(\mathbf{x}_t)]$, add and subtract $\hat{\nabla}_V J(V_t, \bar{\delta}_t; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$, the stochastic quasi-gradient evaluated at $(V_t, \bar{\delta}_t)$ rather than (V_t, z_{t+1}), inside the inner-product term on the right-hand side of (49), to write

$$\begin{aligned}&\|V_{t+1} - V^*\|^2_{\mathcal{H}}\\ &\quad= \|V_t - V^*\|^2_{\mathcal{H}} - 2\alpha_t\langle\hat{\nabla}_V J(V_t, \delta_t; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t), V_t - V^*\rangle_{\mathcal{H}}\\ &\qquad+ 2\epsilon_t\|V_t - V^*\|_{\mathcal{H}} + 2\alpha_t\langle(\gamma\kappa(\mathbf{y}_t, \cdot) - \kappa(\mathbf{x}_t, \cdot))(\bar{\delta}_t - z_{t+1}),\\ &\qquad V_t - V^*\rangle_{\mathcal{H}} + \alpha_t^2\|\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|^2_{\mathcal{H}} \; , \quad (50)\end{aligned}$$

where we substitute in the definitions of $\hat{\nabla}_V J(V_t, \bar{\delta}_t; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ and $\hat{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)$ [cf. (10), (27), respectively] in (50), and cancel out the common regularization term λV_t. We define the directional error associated with difference between the stochastic quasi-gradient and the stochastic gradient as

$$v_t = 2\alpha_t\langle(\gamma\kappa(\mathbf{y}_t, \cdot) - \kappa(\mathbf{x}_t, \cdot))(\bar{\delta}_t - z_{t+1}), V_t - V^*\rangle_{\mathcal{H}} \quad (51)$$

From here, compute the expectation conditional on $\mathcal{F}_t$:

$$\begin{aligned}&\mathbb{E}\left[\|V_{t+1} - V^*\|^2_{\mathcal{H}} \,|\, \mathcal{F}_t\right]\\ &=\|V_t-V^*\|^2_{\mathcal{H}}-2\alpha_t\langle\mathbb{E}\Big[\hat{\nabla}_V J(V_t,\bar{\delta}_t;\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\Big|\mathcal{F}_t\Big],V_t-V^*\rangle_{\mathcal{H}}\\ &\quad+ 2\epsilon_t\|V_t - V^*\|_{\mathcal{H}} + \mathbb{E}\left[v_t \,|\, \mathcal{F}_t\right]\\ &\quad+ \alpha_t^2\mathbb{E}\left[\|\tilde{\nabla}_V J(V_t, z_{t+1}; \mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t)\|^2_{\mathcal{H}} \,\Big|\, \mathcal{F}_t\right] \; . \qquad (52)\end{aligned}$$

Note that the compositional objective $J(V)$ is convex with respect to V, which allows us to write

$$\left\langle\mathbb{E}\Big[\hat{\nabla}_V J(V_t, \bar{\delta}_t; \mathbf{x}_t, \pi(\mathbf{x}_t),\mathbf{y}_t)\Big|\mathcal{F}_t\Big],V_t-V^*\right\rangle_{\mathcal{H}}\geq J(V_t)-J(V^*). (53)$$

Now, we may use Assumption 3 [cf. (33)] regarding the finite conditional moments of the projected stochastic quasi-gradient to the last term in (52) so that it may be replaced by its upper-estimate, which together with (53) simplifies to

$$\begin{aligned}\mathbb{E}\left[\|V_{t+1}-V^*\|^2_{\mathcal{H}}|\mathcal{F}_t\right] &= \|V_t-V^*\|^2_{\mathcal{H}}-2\alpha_t\left[J(V_t)-J(V^*)\right] \quad (54)\\ &\quad+2\epsilon_t\|V_t-V^*\|_{\mathcal{H}}+\alpha_t^2\sigma_V^2+\mathbb{E}\left[v_t|\mathcal{F}_t\right] .\end{aligned}$$

We need to analyze v_t, the directional error associated with using stochastic quasi-gradients rather than stochastic gradients. In doing so, we derive the fact that the sub-optimality $\|V_t - V^*\|$ is intrinsically coupled to the auxiliary sequence $(z_{t+1} - \bar{\delta}_t)$, the focus of Lemma 1(iii). Proceed by applying Cauchy-Schwartz to (51), which allows us to write

$$v_t \leq 2\alpha_t\|\gamma\kappa(\mathbf{y}_t, \cdot)-\kappa(\mathbf{x}_t, \cdot)\|^2_{\mathcal{H}}|z_{t+1}-\bar{\delta}_t|\|V_t-V^*\|_{\mathcal{H}} \quad (55)$$

Note that $2ab \leq \rho a^2 + b^2/\rho$ for $\rho, a, b > 0$, which we apply to (55) with $a = |z_{t+1} - \bar{\delta}_t|$, $b = \alpha_t\|\gamma\kappa(\mathbf{y}_t, \cdot) - \kappa(\mathbf{x}_t, \cdot)\|_{\mathcal{H}}\|V_t - V^*\|_{\mathcal{H}}$, and $\rho = \beta_t$ so that (55) becomes

$$v_t \leq \beta_t(z_{t+1}-\bar{\delta}_t)^2+\frac{\alpha_t^2}{\beta_t}\|\gamma\kappa(\mathbf{y}_t,\cdot)-\kappa(\mathbf{x}_t,\cdot)\|^2_{\mathcal{H}}\|V_t-V^*\|^2_{\mathcal{H}}. \quad (56)$$

The conditional mean of v_t [cf. (51)], using (56), is then

$$\begin{aligned}\mathbb{E}\left[v_t \,\middle|\, \mathcal{F}_t\right] &\leq \beta_t \mathbb{E}\left[(z_{t+1} - \bar{\delta}_t)^2 \,\middle|\, \mathcal{F}_t\right] \\ &\quad + \frac{\alpha_t^2}{\beta_t}\mathbb{E}\left[\|\gamma\kappa(\mathbf{y}_t,\cdot) - \kappa(\mathbf{x}_t,\cdot)\|_{\mathcal{H}}^2 \middle| \mathcal{F}_t\right] \|V_t - V^*\|_{\mathcal{H}}^2 \\ &\leq \beta_t \mathbb{E}\big[(z_{t+1} - \bar{\delta}_t)^2 \big| \mathcal{F}_t\big] + \frac{\alpha_t^2}{\beta_t} G_V^2 \|V_t - V^*\|_{\mathcal{H}}^2 \ ,\end{aligned} \tag{57}$$

where we apply the finite variance property of the functional component of the stochastic gradient [cf. (32)] for the final inequality (57). Substitute (57) into (54) and gather terms:

$$\begin{aligned}&\mathbb{E}\left[\|V_{t+1} - V^*\|_{\mathcal{H}}^2 \,\middle|\, \mathcal{F}_t\right] \\ &\leq \left(1 + \frac{\alpha_t^2}{\beta_t} G_V^2\right) \|V_t - V^*\|_{\mathcal{H}}^2 + 2\epsilon_t \|V_t - V^*\|_{\mathcal{H}} \\ &\quad - 2\alpha_t[J(V_t) - J(V^*)] + \alpha_t^2 \sigma_V^2 + \beta_t \mathbb{E}\left[(z_{t+1} - \bar{\delta}_t)^2 \middle| \mathcal{F}_t\right] ,\end{aligned} \tag{58}$$

which is as stated in Lemma 1(ii). ■

Proof of Lemma 1(iii): This proof is an adaptation of Lemma 2 in Appendix G.1 in the Supplementary Material of [33] to the recursively averaged temporal difference sequence z_t defined in (11). Begin by defining the scalar quantity e_t as the difference of mean temporal 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 difference $\bar{\delta}_t$, plus the difference of mean temporal differences:

$$\begin{aligned}&z_{t+1} - \bar{\delta}_t + e_t \\ &= (1-\beta_t) z_t + \beta_t \delta_t - [(1-\beta_t)\bar{\delta}_t + \beta_t \bar{\delta}_t] + (1-\beta_t)(\bar{\delta}_t - \bar{\delta}_{t-1}) \\ &= (1 - \beta_t)\left(z_t - \bar{\delta}_{t-1}\right) + \beta_t(\delta_t - \bar{\delta}_t)\end{aligned} \tag{59}$$

where we make use of the definition of z_{t+1} in (11), 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 first line of (59), and in the second we gather terms with respect to coefficients $(1 - \beta_t)$ and β_t, and cancel the redundant $\bar{\delta}_t$ term. Now, consider the square of the expression (59), using it's simplification on the right-hand side of the preceding expression

$$\begin{aligned}(z_{t+1} - \bar{\delta}_t + e_t)^2 &= [(1-\beta_t)\left(z_t - \bar{\delta}_{t-1}\right) + \beta_t(\delta_t - \bar{\delta}_t)]^2 \\ &= (1 - \beta_t)^2\left(z_t - \bar{\delta}_{t-1}\right)^2 + \beta_t^2(\delta_t - \bar{\delta}_t)^2 \\ &\quad + 2(1 - \beta_t)\beta_t\left(z_t - \bar{\delta}_{t-1}\right)(\delta_t - \bar{\delta}_t).\end{aligned} \tag{60}$$

where we expand the square to obtain the second line in the previous expression. Now, compute the expectation of (60) conditional on the filtration $\mathcal{F}_t$, which yields

$$\begin{aligned}&\mathbb{E}[(z_{t+1} - \bar{\delta}_t + e_t)^2 \,|\, \mathcal{F}_t] \\ &= (1 - \beta_t)^2\left(z_t - \bar{\delta}_{t-1}\right)^2 + \beta_t^2 \mathbb{E}[(\delta_t - \bar{\delta}_t)^2 \,|\, \mathcal{F}_t] \\ &\quad + 2(1 - \beta_t)\beta_t\left(z_t - \bar{\delta}_{t-1}\right)\mathbb{E}[(\delta_t - \bar{\delta}_t) \,|\, \mathcal{F}_t] \ .\end{aligned} \tag{61}$$

Now we apply the assumption [cf. (31)] that the fact that the temporal difference δ_t is an unbiased estimator for its conditional mean $\bar{\delta}_t$ (so that the last term in the previous expression is null), with finite variance $\mathbb{E}[(\delta_t - \bar{\delta}_t)^2 \,|\, \mathcal{F}_t] \leq \sigma_\delta^2$ (Assumption 3), to write

$$\mathbb{E}[(z_{t+1} - \bar{\delta}_t + e_t)^2 | \mathcal{F}_t] = (1-\beta_t)^2\big(z_t - \bar{\delta}_{t-1}\big)^2 + \beta_t^2 \sigma_\delta^2. \tag{62}$$

We may use the relationship in (62) to 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 $a = z_{t+1} - \bar{\delta}_t + e_t$, $b = -e_t$, and $\rho = \beta_t$. Therefore, we obtain

$$(z_{t+1} - \bar{\delta}_t)^2 \leq (1+\beta_t)(z_{t+1} - \bar{\delta}_t + e_t)^2 + \left(1 + \frac{1}{\beta_t}\right) e_t^2 \ . \tag{63}$$

Now, we use the expected value of (63) in lieu of (62), while gaining a multiplicative factor of $(1 + \beta_t)$ on the right-hand side of (62) plus the error term $(1 + 1/\beta_t)e_t$, yielding

$$\begin{aligned}&\mathbb{E}[(z_{t+1} - \bar{\delta}_t)^2 \,|\, \mathcal{F}_t] \\ &= (1+\beta_t)\Big[(1-\beta_t)^2\big(z_t - \bar{\delta}_{t-1}\big)^2 + \beta_t^2 \sigma_\delta^2\Big] + \left(\frac{1 + \beta_t}{\beta_t}\right) e_t^2.\end{aligned} \tag{64}$$

Apply the fact that $(1-\beta_t^2)(1-\beta_t) \leq (1-\beta_t)$ to the first term in (64) and $(1+\beta_t)\beta_t^2 \leq 2\beta_t^2$ to the second (since $\beta_t \in (0, 1)$) to simplify (64) as

$$\begin{aligned}&\mathbb{E}[(z_{t+1} - \bar{\delta}_t)^2 \,|\, \mathcal{F}_t] \\ &= (1 - \beta_t)\left(z_t - \bar{\delta}_{t-1}\right)^2 + 2\beta_t^2 \sigma_\delta^2 + \left(\frac{1 + \beta_t}{\beta_t}\right) e_t^2.\end{aligned} \tag{65}$$

Now we analyze the term involving e_t, which represents the difference of mean temporal differences. By definition,

$$|e_t| = (1-\beta_t)|(\bar{\delta}_t - \bar{\delta}_{t-1})| \leq (1-\beta_t) L_V \|V_t - V_{t-1}\|_{\mathcal{H}} \tag{66}$$

where we apply the Lipschitz continuity of the conditional average temporal difference $\bar{\delta}_t = \mathbb{E}_{\mathbf{y}_t}[r(\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t) + \gamma V(\mathbf{y}_t) - V(\mathbf{x}_t) \,|\, \mathbf{x}_t, \pi(\mathbf{x}_t)]$ with respect to the value function [cf. (34)] stated in Assumption 4. Substitute the right-hand side of (66) into (65), and simplify the expression in the last term as $(1 - \beta_t^2)/\beta_t \leq 1/\beta_t$ to conclude (41). ■

Lemma 2. *(Coupled Supermartingale Theorem [52][Lemma 6]) Let $\{\xi_k\}$, $\{\zeta_k\}$, $\{u_k\}$, $\{\bar{u}_k\}$, $\{\eta_k\}$, $\{\theta_k\}$, $\{\varepsilon_k\}$, $\{\mu_k\}$, $\{\nu_k\}$ be sequences of nonnegative random variables such that*

$$\mathbb{E}[\xi_{k+1} \,|\, \mathcal{G}_k] \leq (1 + \eta_k)\xi_k - u_k + c\theta_k \zeta_k + \mu_k \ , \tag{67}$$

$$\mathbb{E}[\zeta_{k+1} \,|\, \mathcal{G}_k] \leq (1 - \theta_k)\zeta_k - \bar{u}_k + \varepsilon_k \xi_k + \nu_k \ , \tag{68}$$

where $\mathcal{G}_k = \{\xi_s, \zeta_s, u_s, \bar{u}_s, \eta_s, \theta_s, \varepsilon_s, \mu_s, \nu_s\}_{s=0}^k$ is the filtration, and $c > 0$ is a scalar. Assume the following conditions:

$$\begin{aligned}\sum_{k=0}^{\infty} \eta_k < \infty \ , \quad \sum_{k=0}^{\infty} \varepsilon_k < \infty \ , \\ \sum_{k=0}^{\infty} \mu_k < \infty \ , \ \sum_{k=0}^{\infty} \nu_k < \infty \ ,\end{aligned} \tag{69}$$

almost surely. Then ξ_k and ζ_k converge almost surely to two nonnegative random variables, and we may conclude that

$$\sum_{k=0}^{\infty} u_k < \infty \ , \quad \sum_{k=0}^{\infty} \bar{u}_k < \infty \ , \ \sum_{k=0}^{\infty} \theta_k \zeta_k < \infty \ \ \textit{a. s.} \tag{70}$$

We use Lemma 2 to establish convergence w.p.1 of Algorithm 1 through the expressions derived in Lemma 1.

E. Proof of Theorem 1

We use the relations established in Lemma 1 to construct a coupled supermartingale of the form in Lemma 2 as follows. First, consider the expression (40) for the value function sub-optimality, using approximation budget $\epsilon_t = \alpha_t^2$ and the fact that the value function is bounded in Hilbert norm [cf. (35)] to obtain $\|V_t - V^*\|_{\mathcal{H}} \leq 2K$:

$$\begin{aligned}&\mathbb{E}\left[\|V_{t+1} - V^*\|_{\mathcal{H}}^2 \,\middle|\, \mathcal{F}_t\right]\\&\leq \left(1+\frac{\alpha_t^2}{\beta_t}G_V^2\right)\|V_t - V^*\|_{\mathcal{H}}^2 - 2\alpha_t\left[J(V_t) - J(V^*)\right]\\&\quad+\alpha_t^2(\sigma_V^2+4K)+\beta_t\mathbb{E}\left[(z_{t+1}-\bar{\delta}_t)^2\,\middle|\,\mathcal{F}_t\right]\ .\end{aligned}\tag{71}$$

and then substitute (41) regarding the evolution of z_t with respect to its conditional expectation into (71) to obtain :

$$\begin{aligned}&\mathbb{E}\left[\|V_{t+1} - V^*\|_{\mathcal{H}}^2 \,\middle|\, \mathcal{F}_t\right]\\&\leq \left(1+\frac{\alpha_t^2}{\beta_t}G_V^2\right)\|V_t - V^*\|_{\mathcal{H}}^2 - 2\alpha_t\left[J(V_t) - J(V^*)\right]\\&\quad+\alpha_t^2(\sigma_V^2+4K)+\beta_t(1-\beta_t)(z_t-\bar{\delta}_{t-1})^2\\&\quad+L_V\|V_t - V_{t-1}\|_{\mathcal{H}}^2+2\beta_t^3\sigma_\delta^2\ .\end{aligned}\tag{72}$$

Assume $\beta_t \in (0,1)$. Thus, the right-hand side of (72) yields

$$\begin{aligned}&\mathbb{E}\left[\|V_{t+1} - V^*\|_{\mathcal{H}}^2 \,\middle|\, \mathcal{F}_t\right]\\&\leq \left(1+\frac{\alpha_t^2}{\beta_t}G_V^2\right)\|V_t - V^*\|_{\mathcal{H}}^2 - 2\alpha_t\left[J(V_t) - J(V^*)\right]\\&\quad+\beta_t(z_t-\bar{\delta}_{t-1})^2+\alpha_t^2(\sigma_V^2+4K)\\&\quad+L_V\|V_t - V_{t-1}\|_{\mathcal{H}}^2+2\beta_t^2\sigma_\delta^2\ .\end{aligned}\tag{73}$$

We may identify (73) with the first supermartingale relationship in Lemma 2 [cf. (67)] via the identifications

$$\begin{aligned}&\xi_t = \|V_t - V^*\|_{\mathcal{H}}^2\ ,\eta_t=\frac{\alpha_t^2}{\beta_t}G_V^2\ ,u_t=2\alpha_t[J(V_t)-J(V^*)]\ ,\\&c=1\ ,\qquad\qquad \zeta_t=(z_t-\bar{\delta}_{t-1})^2\ ,\quad \theta_t=\beta_t\ ,\\&\mu_t=\alpha_t^2(\sigma_V^2+4K)+L_V\|V_t-V_{t-1}\|_{\mathcal{H}}^2+2\beta_t^2\sigma_\delta^2\ ,\end{aligned}\tag{74}$$

where $u_t \geq 0$ by the definition of the optimal objective $J(V^*)$. To establish the summability of μ_t, consider Lemma 1(i), which establishes that t $\|V_t - V_{t-1}\|_{\mathcal{H}} \leq \mathcal{O}(\alpha_{t-1}^2)$. Since $\sum_t \alpha_t^2 < \infty$ [cf. (18)], we can sum both sides over all t to conclude the series is finite in conditional expectation:

$$\sum \mathbb{E}[\|V_t - V_{t-1}\|_{\mathcal{H}}\,|\,\mathcal{F}_t] \leq \alpha_{t-1}^2 < \infty.\tag{75}$$

Now, rewrite (75) with total expectation by selecting $\mathcal{F}_0$. Note that since the individual terms $\|V_t - V_{t-1}\|_{\mathcal{H}}^2$ are finite due to the stipulation that the output of KOMP yields finite Hilbert norm value functions, and non-negative by the definition of a norm, we can interchange the expectation (integral) and sum using the Monotone Convergence Theorem to conclude that

$$\mathbb{E}\left[\sum\|V_t - V_{t-1}\|_{\mathcal{H}}\right] < \infty.\tag{76}$$

Thus, $\sum_{t=0}^{\infty}\|V_t - V_{t-1}\|_{\mathcal{H}}^2 < \infty$ w.p.1, implying $\sum_t \mu_t < \infty$.

Now, let's connect the evolution of the auxiliary temporal difference sequence z_t (11) in Lemma 1(iii). In particular, (41) is related to (68) via the identifications:

$$\bar{u}_t = 0\ ,\varepsilon_t = 0\ ,\nu_t = \frac{L_V}{\beta_t}\|V_t - V_{t-1}\|_{\mathcal{H}}^2 + 2\beta_t^2\sigma_\delta^2\ ,\tag{77}$$

with $\zeta_t = (z_t - \bar{\delta}_{t-1})^2$ and $\theta_t = \beta_t$ as in (74). The summability of ν_t follows the following logic: consider the expression $\|V_t - V_{t-1}\|_{\mathcal{H}}^2/\beta_t$ of 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, $\sum_t \|V_t - V_{t-1}\|_{\mathcal{H}}^2/\beta_t < \infty$ almost surely.

Together with the conditions on the step-size sequences α_t and β_t (18), the summability conditions (69) of Lemma 2, the Coupled Supermartingale Theorem, are satisfied, which implies that $\xi_t = \|V_t - V^*\|_{\mathcal{H}}^2$ and $\zeta_t = (z_t - \bar{\delta}_{t-1})^2$ converge to two nonnegative random variables almost surely, and that:

$$\sum_t \alpha_t[J(V_t) - J(V^*)] < \infty\ ,\quad \sum_t \beta_t(z_{t+1} - \bar{\delta}_t)^2 < \infty\ ,\tag{78}$$

almost surely. The non-summability of the step-size sequences α_t and β_t (18) allows us to conclude that:

$$\liminf_{t\to\infty} J(V_t) = J(V^*)\ ,\quad \liminf_{t\to\infty}(z_{t+1} - \bar{\delta}_t)^2 = 0\ ,\tag{79}$$

almost surely, and that $\|V_t - V^*\|_{\mathcal{H}}^2$ converges to a nonnegative random variable with probability 1, as does $(z_{t+1} - \bar{\delta}_t)^2$. We proceed to show that the entire sequence must converge. The rest of this proof is analogous to [33], but is repeated here for completeness. Let Ω_{V^*} be the collection of sample paths such that $\Omega_{V^*} = \{\mathbf{y} : \lim_t \|V_t(\mathbf{y}) - V^*\| \text{ exists }\}$. Here we use the notation not that the value function is evaluated at state $\mathbf{y}$ but instead is a function of random variable $\mathbf{y}$. We just established above that $\mathbb{P}(\Omega_{V^*}) = 1$ for any $V^* \in \mathcal{H}$. To prove that any limiting value function is optimal, we need to establish that $\cap_{V^*\in\mathcal{H}}\Omega_{V^*}$ is measurable and $\mathbb{P}(\cap_{V^*\in\mathcal{H}}\Omega_{V^*}) = 1$.

To do so, note that since J is convex, the set of minimizers of J, denoted as $\mathcal{H}^* \subset \mathcal{H}$, is separable, and has a countably dense subset $\mathcal{H}_Q^*$. Thus the probability of divergence for some $V^* \in \mathcal{H}_Q^*$ is the probability of a union of countably many sets, each having null probability. Therefore, we may write

$$\mathbb{P}\left(\cap_{\mathcal{H}_Q^*}\Omega_{V^*}\right) = 1 - \mathbb{P}\left(\cup_{\mathcal{H}_Q^*}\Omega_{V^*}^c\right) \geq 1 - \sum_{V^*\in\mathcal{H}_Q^*}\mathbb{P}\left(\Omega_{V^*}^c\right) = 1\tag{80}$$

by simple application of De Morgan's Law and Boole's inequality. Then consider any $\tilde{V} \in \mathcal{H}^\star$ which is the limit of a sequence of optimal value functions $\{\tilde{V}_k\}_{k=1}^{\infty} \subset \mathcal{H}^\star$. We can prove that $\|\tilde{V}_t(\mathbf{y}) - \tilde{V}\|$ is convergent provided that $\|\tilde{V}_t(\mathbf{y}) - \tilde{V}_k\|$ is convergent for all k. Note that

$$\begin{aligned}&\|V_t(\mathbf{y}) - \tilde{V}_k\|_{\mathcal{H}} - \|\tilde{V}_k - \tilde{V}\|_{\mathcal{H}}\\&\qquad\leq \|V_t(\mathbf{y}) - \tilde{V}\|_{\mathcal{H}}\\&\qquad\leq \|V_t(\omega) - \tilde{V}_k\|_{\mathcal{H}} + \|\tilde{V}_k - \tilde{V}\|_{\mathcal{H}}\ .\end{aligned}\tag{81}$$

Since $\|V_t(\mathbf{y})-\tilde{V}_k\|_{\mathcal{H}}$ has a limit, take $t\to\infty$ in (81), yielding:

$$\begin{aligned}\lim_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}_k\|_{\mathcal{H}}-\|\tilde{V}_k-\tilde{V}\|_{\mathcal{H}}&\le\liminf_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}\\ &\le\limsup_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}} \qquad (82)\\ &\le\lim_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}_k\|_{\mathcal{H}}+\|\tilde{V}_k-\tilde{V}\|_{\mathcal{H}}\ ,\end{aligned}$$

which, by subtracting $\liminf_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}$ from both sides in (82), cancelling the common $\lim_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}_k\|_{\mathcal{H}}$, and combining terms, allows us to write

$$\limsup_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}-\liminf_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}\le 2\|\tilde{V}_k-\tilde{V}\|_{\mathcal{H}}. \qquad (83)$$

Take $k\to\infty$ in (83), for which $\|\tilde{V}_t-\tilde{V}\|_{\mathcal{H}}\to 0$, hence

$$\limsup_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}=\liminf_{t\to\infty}\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}\ , \qquad (84)$$

and therefore $\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}$ has a limit, so $\mathbf{y}\in\Omega_{\tilde{V}^*}$, and therefore $\cap_{\mathcal{H}_Q^*}\Omega_{V^*}\subset\Omega_{\tilde{V}}$. Consequently, $\mathbb{P}\left(\cap_{\mathcal{H}_Q^*}\Omega_{V^*}\right)=1$. As a result, we have $(\cap_{\mathcal{H}^*}\Omega_{V^*})^c\subset\left(\cap_{\mathcal{H}_Q^*}\Omega_{V^*}\right)^c$, both of which are measurable and have null probability: $\mathbb{P}\left((\cap_{\mathcal{H}^*}\Omega_{V^*})^c\right)\le\mathbb{P}\left((\cap_{\mathcal{H}_Q^*}\Omega_{V^*})^c\right)=0$. Thus, $(\cap_{\mathcal{H}^*}\Omega_{V^*})$ is measurable and occurs with probability 1. Put another way, $\|V_t-\tilde{V}\|_{\mathcal{H}}$ is convergent for all $\tilde{V}\in\mathcal{H}^*$ with probability 1.

Now, we can use this fact together with (79), namely, $\liminf_{t\to\infty}J(V_t)=J(V^*)$, to establish that V_t converges to the minimizer of $J(V)$ a.s. To do so, let $V^*\in\mathcal{H}^*$ the set of optimizers of J. Since $\|V_t(\mathbf{y})-V^*\|_{\mathcal{H}}$ converges, it is bounded. Then, $\{V_t(\mathbf{y})\}$ must have a limit point $\tilde{V}$ being an optimal solution, $J(\tilde{V})=J^*$ with $\tilde{V}\in\mathcal{H}^*$, by the continuity of J. Since $\omega\in\cap\mathcal{H}^*\Omega_{V^*}\subset\Omega_{\tilde{V}}$, $\{\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}\}$ is a convergent sequence whose limit is null. Thus, $\|V_t(\mathbf{y})-\tilde{V}\|_{\mathcal{H}}\to 0$, so $V_t(\mathbf{y})\to\tilde{V}$ on this sample path. $\tilde{V}$ is a random variable dependent on the sample path, parameterized by $\mathbf{y}$. The set of all such sample paths has prob. 1, so that V_t converges to a random point in $\mathcal{H}^*$. ■

F. Proof of Theorem 2

Before analyzing the mean convergence behavior of the value function, we consider the mean sub-optimality of the auxiliary variable z_t with respect to the conditional mean of the temporal difference $\bar{\delta}_t$. To do so, compute the total expectation of Lemma 1(iii), stated as

$$\begin{aligned}&\mathbb{E}\left[(z_{t+1}-\bar{\delta}_t)^2\right] \qquad (85)\\ &\le(1-\beta)\mathbb{E}\left[(z_t-\bar{\delta}_{t-1})^2\right]+\frac{L_V}{\beta}\mathbb{E}\left[\|V_t-V_{t-1}\|_{\mathcal{H}}^2\right]+2\beta^2\sigma_\delta^2\ ,\end{aligned}$$

where we have substituted in constant learning rate $\beta_t=\beta$ in (85). The total expectation of Lemma 1(i) regarding $\|V_t-V_{t-1}\|_{\mathcal{H}}^2$, the difference of value functions in Hilbert-norm, may be substituted into (85), with constant step-size $\alpha_t=\alpha$ and compression budgets $\epsilon_t=\epsilon$ to obtain

$$\begin{aligned}&\mathbb{E}\left[(z_{t+1}-\bar{\delta}_t)^2\right]\\ &\quad\le(1-\beta)\mathbb{E}\left[(z_t-\bar{\delta}_{t-1})^2\right]\\ &\qquad+\frac{2L_V}{\beta}\left[\alpha^2(G_\delta^2G_V^2+\lambda^2K^2)+\epsilon^2\right]+2\beta^2\sigma_\delta^2\ , \qquad (86)\end{aligned}$$

Observe that (86) gives a relationship between the sequence $\mathbb{E}\left[(z_{t+1}-\bar{\delta}_t)^2\right]$ and its value at the previous iterate. We can substitute $t+1$ by t in (86) to write

$$\begin{aligned}\mathbb{E}\left[(z_t-\bar{\delta}_{t-1})^2\right]\le(1-\beta)\mathbb{E}\left[(z_{t-1}-\bar{\delta}_{t-2})^2\right]+\frac{2L_V}{\beta} \qquad (87)\\ \times\left[\alpha^2(G_\delta^2G_V^2+\lambda^2K^2)+\epsilon^2\right]+2\beta^2\sigma_\delta^2\ ,\end{aligned}$$

Substituting (87) into the right-hand side of (86) yields

$$\begin{aligned}\mathbb{E}\left[(z_{t+1}-\bar{\delta}_t)^2\right]\le(1-\beta)^2\mathbb{E}\left[(z_{t-1}-\bar{\delta}_{t-2})^2\right]+\left[1+(1-\beta)\right] \qquad (88)\\ \times\left\{\frac{2L_V}{\beta}\left[\alpha^2(G_\delta^2G_V^2+\lambda^2K^2)+\epsilon^2\right]+2\beta^2\sigma_\delta^2\right\}.\end{aligned}$$

We can recursively apply the previous two steps backwards in time to the initialization to obtain

$$\begin{aligned}\mathbb{E}\left[(z_{t+1}-\bar{\delta}_t)^2\right]\le(1-\beta)^{t+1}(z_0-\bar{\delta}_{-1})^2+\sum_{u=0}^{t}(1-\beta)^u\left\{\frac{2L_V}{\beta}\right.\\ \left.\times\left[\alpha^2(G_\delta^2G_V^2+\lambda^2K^2)+\epsilon^2\right]+2\beta^2\sigma_\delta^2\right\}, \qquad (89)\end{aligned}$$

In (89), the first term on the left-hand side vanishes due to the initialization $z_0=0$ and the convention $\delta_{-1}=0$. Moreover, the finite geometric sum may be evaluated, provided $\beta<1$, as $\sum_{u=0}^{t}(1-\beta)^u=[1-(1-\beta)^t]/\beta$. The numerator in this simplification is strictly less than unit, which means that the right-hand side of (89) simplifies to

$$\begin{aligned}\mathbb{E}\left[(z_{t+1}-\bar{\delta}_t)^2\right]&\le\frac{2L_V}{\beta^2}\left[\alpha^2(G_\delta^2G_V^2+\lambda^2K^2)+\epsilon^2\right]+2\beta\sigma_\delta^2\\ &=\mathcal{O}\left(\frac{\alpha^2+\epsilon^2}{\beta^2}+\beta\right) \qquad (90)\end{aligned}$$

With this relationship established for the auxiliary sequence z_t, we shift gears to addressing the evolution of the value function sub-optimality $\|V_t-V^*\|_{\mathcal{H}}$ in expectation. Begin by using the fact that the Hilbert-norm regularizer $(\lambda/2)\|V\|_{\mathcal{H}}^2$ in (8) implies the objective $J(V)$ is strongly convex, i.e.

$$\frac{\lambda}{2}\|V_t-V^*\|_{\mathcal{H}}^2\le J(V_t)-V(V^*)\ , \qquad (91)$$

together with the expression in Lemma 1(ii) regarding the value function sub-optimality, assuming constant learning rates and compression budget, i.e. $\alpha_t=\alpha,\beta_t=\beta,\epsilon_t=\epsilon$, to write

$$\begin{aligned}&\mathbb{E}\left[\|V_{t+1}-V^*\|_{\mathcal{H}}^2\,\middle|\,\mathcal{F}_t\right]\\ &\quad\le\left(1+\frac{\alpha^2}{\beta}G_V^2-\alpha\lambda\right)\|V_t-V^*\|_{\mathcal{H}}^2+2\epsilon\|V_t-V^*\|_{\mathcal{H}}\\ &\qquad+\alpha^2\sigma_V^2+\beta\mathbb{E}\left[(z_{t+1}-\bar{\delta}_t)^2\,\middle|\,\mathcal{F}_t\right]\ . \qquad (92)\end{aligned}$$

Consider the total expectation of (92) with choice of compression budget $\epsilon=C\alpha^2$ for some arbitrary constant $C>0$, the fact that $\|V_t-V^*\|_{\mathcal{H}}\le 2K$, apply (90) to the last term on the right-hand side of (92), and substitute in regularizer $\lambda = G_V^2\alpha/\beta + \lambda_0$ for $\lambda_0 < 1$ to obtain:

$$\mathbb{E}\left[\|V_{t+1} - V^*\|_{\mathcal{H}}^2\right] \leq (1-\lambda_0)\,\mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] + \alpha^2(\sigma_V^2 + 4CK) + 2\beta^2\sigma_\delta^2 + \frac{2L_V}{\beta}\big[\alpha^2(G_\delta^2 G_V^2 + \lambda^2 K^2) + C^2\alpha^4\big]. \tag{93}$$

To establish that $\liminf \|V_t - V^*\|_{\mathcal{H}}^2$ is a finite constant determined by λ_0 and the constant terms on the right-hand side of (93), which we define as $R := \alpha^2(\sigma_V^2 + 4CK) + 2\beta^2\sigma_\delta^2 + \frac{2L_V}{\beta}\big[\alpha^2(G_\delta^2 G_V^2 + \lambda^2 K^2) + C^2\alpha^4\big]$, suppose that it is not, i.e., that the following holds true:

$$\liminf_t \mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] > \frac{R}{\lambda_0} \tag{94}$$

Then there exists some time index $t_0 < \infty$ and some $\delta > 0$ such that

$$\mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] > \frac{R}{\lambda_0} + \delta \tag{95}$$

for all $t \geq t_0$. Note that (95) may be rearranged to equivalently be stated as

$$\lambda_0\mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] - \lambda_0\delta > R \tag{96}$$

Let's substitute upper-bound for R stated in (96) into (93):

$$\begin{aligned}\mathbb{E}\left[\|V_{t+1} - V^*\|_{\mathcal{H}}^2\right] &\leq (1-\lambda_0)\,\mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] + R \\ &< \mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] - \lambda_0\delta \\ &\leq \mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right]\end{aligned} \tag{97}$$

where we have cancelled a common factor of $\lambda_0\mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right]$ from the right-hand side, and upper-estimated $-\lambda_0\delta$ by null. Therefore, under the hypothesis that $\liminf_t \mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] > R/\lambda_0$, by (97), $\mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right]$ decreases monotonically to null. This is a contradiction. Therefore, we must have that the hypothesis (94) is false, and hence

$$\begin{aligned}\liminf_{t\to\infty} \mathbb{E}\left[\|V_t - V^*\|_{\mathcal{H}}^2\right] &= R \\ &= \mathcal{O}\left(\alpha^2 + \beta^2 + \frac{\alpha^2}{\beta}\left[1 + \alpha^2 + \frac{\alpha}{\beta} + \frac{\alpha^2}{\beta^2}\right]\right)\ .\end{aligned} \tag{98}$$

When $\alpha = \beta$, the posynomial of the learning rates on the right-hand side of (98) simplifies to be $\mathcal{O}(\alpha + \alpha^2 + \alpha^3) = \mathcal{O}(\alpha)$ for $\alpha \in (0,1)$ as stated in (20) (Theorem 2).

G. Proof of Corollary 1

We prove Corollary 1: In Theorem 3 of [32][Appendix D.1], it is established for a nonparametric stochastic program without any compositional structure that the effect of sparse subspace projections on the functional stochastic gradient sequence in an RKHS is to yield a function sequence of finite model order, provided a constant algorithm step-size and compression budget are used. The proof of Corollary 1 is nearly identical: the same projection operator is used and the same compactness properties of the state and action spaces apply. The only point of departure is that a distinct deterministic bound is needed on the functional stochastic quasi-gradient for all $\{\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t\}$, i.e., to apply the reasoning following equations (74) in [32][Appendix D.1], we require the existence of a deterministic constant D such that $|[\gamma\kappa(\mathbf{y}_t,\cdot) - \kappa(\mathbf{x}_t,\cdot)]z_{t+1}| \leq D$ for all $\{\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t\}$. We establish such an upper-estimate. To do so, we first establish that the auxiliary sequence z_t stated in (11) is bounded, i.e.

Proposition 2. *The auxiliary sequence z_t [cf.* (11)*] is upper-bounded when used with constant step-size $\beta_t = \beta$:*

$$|z_t| = (\gamma+1)K + R_{\max} \text{ for all } t \tag{99}$$

Proof: We pursue a proof by induction. First, the base case: with $V_0 = 0$, we have $|z_1| \leq \beta R_{\max} \leq (\gamma+1)K + R_{\max}$ making use of the bound on V_t for all t in (35) and the fact that the step-size is less than unit. Now, the induction step: assume the prior bound holds for z_u for $u \leq t$. Write for z_{t+1}

$$|z_{t+1}| = (1-\beta)|z_t| + \beta|\delta_t| \leq (\gamma+1)K + R_{\max} \tag{100}$$

where in the last inequality we apply the induction hypothesis together with the upper-estimate on the temporal difference $\delta_t \leq (\gamma+1)K + R_{\max}$. ■

By making use of Proposition 2 together with the bound on the reproducing kernel map (Assumption 2), we have the following uniform deterministic bound:

$$\begin{aligned}|[\gamma\kappa(\mathbf{y}_t,\cdot) - \kappa(\mathbf{x}_t,\cdot)]z_{t+1}| &\leq X(\gamma+1)[(\gamma+1)K + R_{\max}] \\ &:= D \text{ for all } \{\mathbf{x}_t, \pi(\mathbf{x}_t), \mathbf{y}_t\}\end{aligned} \tag{101}$$

Then, we may apply the same reasoning as that of Appendix D.1 of [32] to conclude that the number of Euclidean balls of radius $d = \epsilon/D$ needed to cover the space $\phi(\mathcal{X}) = \kappa(\mathcal{X},\cdot)$ is finite, where ϵ is a constant as in (19). See [57] for further details. Therefore, for Algorithm 1, there exists a finite $M^\infty < \infty$ such that the model order $M_t \leq M^\infty$ for all t.