Policy Gradient and Actor–Critic Learning in Continuous Time and Space: Theory and Algorithms

Yanwei Jia yj2650@columbia.edu
Department of Industrial Engineering and Operations Research
Columbia University
New York, NY 10027, USA

Xun Yu Zhou xz2574@columbia.edu
Department of Industrial Engineering and Operations Research &
The Data Science Institute
Columbia University
New York, NY 10027, USA

Abstract

We study policy gradient (PG) for reinforcement learning in continuous time and space under the regularized exploratory formulation developed by Wang et al. (2020). We represent the gradient of the value function with respect to a given parameterized stochastic policy as the expected integration of an auxiliary running reward function that can be evaluated using samples and the current value function. This effectively turns PG into a policy evaluation (PE) problem, enabling us to apply the martingale approach recently developed by Jia and Zhou (2021) for PE to solve our PG problem. Based on this analysis, we propose two types of the actor–critic algorithms for RL, where we learn and update value functions and policies simultaneously and alternatingly. The first type is based directly on the aforementioned representation which involves future trajectories and hence is offline. The second type, designed for online learning, employs the first-order condition of the policy gradient and turns it into martingale orthogonality conditions. These conditions are then incorporated using stochastic approximation when updating policies. Finally, we demonstrate the algorithms by simulations in two concrete examples.

Keywords: Reinforcement learning, continuous time and space, policy gradient, policy evaluation, actor–critic algorithms, martingale.

1. Introduction

The essence of reinforcement learning (RL) is "trial and error": to repeatedly try a policy for actions, receive and evaluate reward signals, and improve the policy. This manifests three key components of RL: 1) **exploration** with *stochastic* policies - to broaden search space via randomization; 2) **policy evaluation** - to evaluate the value function of a current policy; and 3) **policy improvement** - to improve the current policy. Numerous algorithms have been proposed in the RL literature, which can be generally categorized into three types: critic-only, actor-only, and actor–critic. Here, an *actor* refers to a policy that governs the actions, and a *critic* refers to the value function that evaluates the performance of a policy. The *critic-only* approach learns a value function to compare the estimated outcomes of different actions and selects the best one in accordance with the current value function. The *actor-only* approach acts directly without learning the expected outcomes of different policies. The *actor–critic* approach uses simultaneously an actor to improve the policy for generating actions given the current state of the environment and a critic to judge the selected policy and to guide improving the actor. See Sutton and Barto (2018) and the references therein for extensive discussions on these methods.

All these algorithms and indeed the general RL study have been hitherto predominantly limited to *discrete-time* Markov decision processes (MDPs). It remains a largely uncharted territory to study RL in continuous time with continuous state and action spaces. The few existing papers on RL in the continuous setting are mostly restricted to *deterministic* systems; see for example Baird (1993); Doya (2000); Frémaux et al. (2013); Vamvoudakis and Lewis (2010); Lee and Sutton (2021) where there are no environmental noises. In real life, however, examples abound in which an agent can or actually *needs* to interact with a random environment at ultra-high frequency or outright continuously, e.g., high-frequency stock trading, autonomous driving, and robots navigation.

The RL research for continuous-time diffusion processes started only recently. Wang et al. (2020) propose an entropy-regularized stochastic relaxed control framework to study RL in continuous time and space, and derive Boltzmann distributions as the generally optimal stochastic policies for exploring environment and generating actions. In particular, when the problem is linear–quadratic (LQ), namely the dynamic is linear and the reward is quadratic in state and action, the optimal policy specializes to Gaussian distributions. Extensions and applications of this work include Wang and Zhou (2020); Dai et al. (2020); Guo et al. (2020); Gao et al. (2020).

While Wang et al. (2020) address the first component of RL – exploration - for the continuous setting, Jia and Zhou (2021) investigate the second component, namely policy evaluation (PE), aiming at establishing a theoretical foundation for PE in continuous time and space. They show that PE is theoretically equivalent to maintaining the martingale condition of a specifically defined stochastic process, based on which they propose several online and offline PE algorithms. These algorithms have discrete-time counterparts, such as gradient Monte Carlo, TD(λ), and GTD, that scatter around in the MDP RL literature. Therefore, through the "martingale lens", Jia and Zhou (2021) not only devise new PE algorithms for the continuous case but also interpret and unify many classical algorithms originally designed for MDPs.

The current paper is a continuation of Wang et al. (2020) and Jia and Zhou (2021), dealing with the third component of RL – policy improvement – in the continuous setting under stochastic policies and, thereby, completing the full procedure for typical RL tasks. Note that Wang and Zhou (2020) put forth a policy improvement theorem, for the special case of a continuous-time mean–variance portfolio selection problem, in which a new policy selected by maximizing the Hamiltonian of the currently leaned value function is proved to achieve a better objective value than the current policy. This method, akin to Q-learning for MDPs, has a drawback in requiring the functional form of the Hamiltonian which in turn involves the knowledge of the environment. Moreover, even if the Hamiltonian is known, maximizing a potentially complex function in high dimensions is a computationally demanding, or indeed daunting, task.

In this paper, we take a different approach – that of *policy gradient* (PG) – which optimizes the value function over a parameterized family of policies. This approach has at least two advantages. First, selecting actions does not involve any maximization; instead, actions are sampled from a known parametric distribution. Second, approximating policies directly facilitates more efficient learning, if one has prior knowledge or intuition about the classes of potentially optimal policies (e.g. Gaussian distributions) leading to fewer parameters of the parametric family to be learned.

PG as a general sub-method of RL has a long history that can be traced back to Aleksandrov et al. (1968); Glynn (1990); Williams (1992); Barto et al. (1983); see also Bhatnagar et al. (2009) for more literature review and references therein. PG theorems specifically for MDPs are established in Sutton et al. (2000a) and Marbach and Tsitsiklis (2001). Deterministic policy gradient algorithms for semi-MDPs (with discrete time and continuous action space) are developed in Silver et al. (2014) and later extended to incorporate deep neural networks in Lillicrap et al. (2015). Empirically, however, such algorithms tend to be unstable (Duan et al., 2016). Recent study has focused on stochastic policies with possible entropy regularizers, also known as the *softmax* method; see for example Mnih et al. (2016); Schulman et al. (2017a,b); Haarnoja et al. (2018).

PG updates and improves policies along the gradient ascent direction, and is often carried out simultaneously and alternatingly with PE. The resulting algorithms for RL, therefore, are essentially actor–critic (AC) ones. Such methods have been successful in many real-world applications, notably AlphaGo (Silver et al., 2017) and dexterous hand manipulation (Haarnoja et al., 2018). But then, again, most PG and AC algorithms have been developed for discrete-time MDPs, many of them in heuristic and *ad hoc* manners. It remains significant questions to find their continuous-time counterparts and, more importantly, to lay an overarching theoretical underpinning of them.

This paper aims to answer these questions, by studying PG for a general problem in continuous time and space, based on which we develop model-free AC algorithms for RL, covering both episodic and continuing, and both online and offline tasks. As its predecessors, Wang et al. (2020) and Jia and Zhou (2021), we develop theory in continuous time and discretize time only at the final algorithmic implementation stage, instead of discretizing time upfront and applying the existing MDP results. Specifically, we carry out our analysis in the stochastic relaxed control framework of Wang et al. (2020) involving distribution-valued stochastic policies. As such, it is necessary to first extend the PE theory of Jia and Zhou (2021), including the martingale characterization and the resulting methods of martingale loss function and martingale orthogonality conditions, from deterministic policies to stochastic ones. This extension is technically non-trivial. Our main contributions, however, are a thorough analysis of the PG and the resulting AC algorithms. More precisely, we deduce the representation of the gradient of the current value function with respect to a parameterized (stochastic) policy. This representation turns out to have the *same form* as the value function in the PE step, effectively turning PG into an auxiliary PE problem. However, a subtle difficulty is that the corresponding "auxiliary" reward depends on the Hamiltonian and hence on the functional forms of the system dynamics. We solve this difficulty by integration by parts and Itô's formula, transforming the representation into the expected integration of functions that can be evaluated using samples along with the current value function approximator.

The aforementioned representation is forward-looking, namely, it is the conditional expectation of a term involving *future* states. Hence it is suitable for offline learning only. For online learning, we employ the first-order condition of the policy gradient and turn it into martingale orthogonality conditions. These conditions are then incorporated using stochastic approximation when updating policies. Finally, combining the newly developed PG methods in this paper and the PE methods in Jia and Zhou (2021), we propose several AC algorithms for episodic and continuing/ergodic tasks.

Within the continuous-time stochastic relaxed control framework, there are several studies involving updating policies. For example, Wang and Zhou (2020) consider mean–variance portfolio selection and update policies by maximizing the Hamiltonian. As mentioned earlier, this requires knowledge about the market and hence the method is essentially model-based. Dai et al. (2020) address the time-inconsistency issue and focus on learning equilibrium policies. Guo et al. (2020) study multi-agent RL by solving an LQ mean-field game. These two papers rely on differentiating with respect to the policy hence are both model-based methods. In contrast, the present paper provides general model-free (up to the underlying dynamics being diffusion processes) AC algorithms that can be applied in all the above problems. In particular, we apply our algorithms to the mean–variance portfolio selection problem in Wang and Zhou (2020) and show they outperform significantly.

The rest of the paper proceeds as follows. In Section 2, we review Wang et al. (2020)'s entropy-regularized, exploratory formulation for RL in continuous time and space, and put forth an equivalent formulation convenient for the subsequent analysis. In Section 3, we develop a theory for PG, based on which we present general AC algorithms. Section 4 is devoted to an extension to ergodic tasks. We demonstrate our algorithms by simulation with two concrete examples in Section 5. Finally, Section 6 concludes. Proofs of the results are relegated to the appendix.

2. Problem Formulation and Preliminaries

Throughout this paper, by convention all vectors are *column* vectors unless otherwise specified, and $\mathbb{R}^k$ is the space of all k-dimensional vectors (hence $k\times 1$ matrices). Let A and B be two matrices of the same size. We denote by $A\circ B$ the inner product between A and B, by $|A|$ the Eulidean/Frobenius norm of A, and write $A^2 := AA^\top$, where $A^\top$ is A's transpose. For a positive semidefinite matrix A, we write $\sqrt{A} = UD^{1/2}V^\top$, where $A = UDV^\top$ is its singular value decomposition with U, V two orthogonal matrices and D a diagonal matrix, and $D^{1/2}$ is the diagonal matrix whose entries are the square root of those of D. We use $f = f(\cdot)$ to denote the *function* f, and $f(x)$ to denote the *function value* of f at x. For any stochastic process $X = \{X_s,\, s\geq 0\}$, we denote by $\{\mathcal{F}_s^X\}_{s\geq 0}$ the natural filtration generated by X. Finally, for any filtration $\mathcal{G} = \{\mathcal{G}_s\}_{s\geq 0}$ and any semi-martingale $Y = \{Y_s,\, s\geq 0\}$, we denote

$$L^2_{\mathcal{G}}([0,T];Y) = \left\{\kappa = \{\kappa_t, 0\leq t\leq T\} :\ \kappa \text{ is } \mathcal{G}_t\text{-progressively measurable and } \mathbb{E}\int_0^T |\kappa_t|^2 \mathrm{d}\langle Y\rangle_t < \infty\right\}$$

which is a Hilbert space with the L^2-norm $||\kappa||_{L^2} = \left(\mathbb{E}\int_0^T \kappa_t^2 \mathrm{d}\langle Y\rangle_t\right)^{\frac{1}{2}}$, where $\langle\cdot\rangle$ is the quadratic variation of a given process.

Let d, n be given positive integers, $T > 0$, and $b : [0,T]\times\mathbb{R}^d\times\mathcal{A}\mapsto\mathbb{R}^d$ and $\sigma : [0,T]\times\mathbb{R}^d\times\mathcal{A}\mapsto\mathbb{R}^{d\times n}$ be given functions, where $\mathcal{A}$ is the action set. The classical stochastic control problem is to control the *state* (or *feature*) dynamics governed by a stochastic differential equation (SDE), defined on a filtered probability space $\big(\Omega, \mathcal{F}, \mathbb{P}^W; \{\mathcal{F}_s^W\}_{s\geq 0}\big)$ along with a standard n-dimensional Brownian motion $W = \{W_s, s \geq 0\}$:

$$\mathrm{d}X_s^a = b(s, X_s^a, a_s)\mathrm{d}s + \sigma(s, X_s^a, a_s)\mathrm{d}W_s,\ \ s \in [0, T], \tag{1}$$

where a_s stands for the agent's action (control) at time s. The goal of stochastic control is, for each initial time-state pair (t, x) of (1), to find the optimal $\{\mathcal{F}_s^W\}_{s\geq 0}$-progressively measurable (continuous) sequence of actions $a = \{a_s, t \leq s \leq T\}$ – also called the optimal strategy – that maximizes the expected total reward:

$$\mathbb{E}^{\mathbb{P}^W}\left[\int_t^T e^{-\beta(s-t)} r(s, X_s^a, a_s)\mathrm{d}s + e^{-\beta(T-t)} h(X_T^a)\Big| X_t^a = x\right]. \tag{2}$$

Note in the above the state process $X^a = \{X_s^a, t \leq s \leq T\}$ also depends on (t, x). However, to ease notation, here (and similarly in the sequel) we use X^a instead of $X^{t,x,a} = \{X_s^{t,x,a}, t \leq s \leq T\}$ to denote the solution to SDE (1) with initial condition $X_t^a = x$ whenever no ambiguity may arise.

Let $\mathcal{L}^a$ be the *infinitesimal generator* associated with the diffusion process governed by (1):

$$\mathcal{L}^a\varphi(t, x) := \frac{\partial\varphi}{\partial t}(t, x) + b\big(t, x, a\big) \circ \frac{\partial\varphi}{\partial x}(t, x) + \frac{1}{2}\sigma^2\big(t, x, a\big) \circ \frac{\partial^2\varphi}{\partial x^2}(t, x),\ \ a \in \mathcal{A},$$

where $\frac{\partial\varphi}{\partial x} \in \mathbb{R}^d$ is the gradient, and $\frac{\partial^2\varphi}{\partial x^2} \in \mathbb{R}^{d\times d}$ is the Hessian. We make the following assumption to ensure theoretically the well-posedness of the stochastic control problem (1)–(2).

Assumption 1 *The following conditions for the state dynamics and reward functions hold true:*

(i) b, σ, r, h *are all continuous functions in their respective arguments;*

(ii) b, σ *are uniformly Lipschitz continuous in* x*, i.e., for* $\varphi = b,\ \sigma$*, there exists a constant* $C > 0$ *such that*

$$|\varphi(t, x, a) - \varphi(t, x', a)| \leq C|x - x'|,\ \ \forall (t, a) \in [0, T] \times \mathcal{A},\ \forall x, x' \in \mathbb{R}^d;$$

(iii) b, σ *have linear growth in* x*, i.e., for* $\varphi = b,\ \sigma$*, there exists a constant* $C > 0$ *such that*

$$|\varphi(t, x, a)| \leq C(1 + |x|),\ \ \forall (t, x, a) \in [0, T] \times \mathbb{R}^d \times \mathcal{A};$$

(iv) r *and* h *have polynomial growth in* (x, a) *and* x *respectively, i.e., there exists a constant* $C > 0$ *and* $\mu \geq 1$ *such that*

$$|r(t, x, a)| \leq C(1 + |x|^\mu + |a|^\mu),\ \ |h(x)| \leq C(1 + |x|^\mu,\ \forall (t, x, a) \in [0, T] \times \mathbb{R}^d \times \mathcal{A}.$$

Classical model-based stochastic control theory has been well developed (e.g., Fleming and Soner, 2006 and Yong and Zhou, 1999) to solve the above problem, under the premise that the functional forms of b, σ, r, h are all given and known. In the RL setting, however, the agent does not have this knowledge of the environment. Instead, what she can do is "trial and error" – to try a (continuous) sequence of actions $a = \{a_s, t \le s \le T\}$, observe the corresponding state process $X^a = \{X^a_s, t \le s \le T\}$ and collect both a stream of discounted running rewards $\{e^{-\beta(s-t)} r(s, X^a_s, a_s), t \le s \le T\}$ and a discounted, end-of-period lump-sum reward $e^{-\beta(T-t)} h(X^a_T)$ where β is a given, known discount factor. In the offline setting, the agent can repeatedly try different sequences of actions over the same time period $[0, T]$ and record the corresponding state processes and payoffs. In the online setting, the agent updates the actions as she goes, based on all the up-to-date historical observations.

A critical question is how to *generate* these trial-and-error sequences of actions. The idea is *randomization*, namely, the agent employs a *stochastic policy*, which is a probability distribution on the action space, to produce actions according to the current time–state pair. It is important to note that this randomization itself is *independent* of the underlying Brownian motion W, the random source of the original control problem that stands for the environmental noise. Wang et al. (2020) formulate an RL problem in continuous time and space, incorporating distribution-valued stochastic policies with an entropy regularizer to account for the tradeoff between exploration and exploitation. Specifically, assume the probability space is rich enough to support a random variable Z that is uniformly distributed on $[0, 1]$ and independent of W. We then expand the original filtered probability space to $(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_s\}_{s\ge 0})$ where $\mathcal{F}_s = \mathcal{F}^W_s \vee \sigma(Z)$ and $\mathbb{P}$ is now the probability measure on $\mathcal{F}_T$.[1] Let $\boldsymbol{\pi} : (t, x) \in [0, T] \times \mathbb{R}^d \mapsto \boldsymbol{\pi}(\cdot|t, x) \in \mathcal{P}(\mathcal{A})$ be a given (feedback) policy, where $\mathcal{P}(\mathcal{A})$ is a suitable collection of probability density functions (pdfs).[2] At each time s, an action a_s is generated or sampled from the distribution $\boldsymbol{\pi}(\cdot|s, X_s)$.

Given a stochastic policy $\boldsymbol{\pi}$, an initial time–state pair (t, x), and an $\{\mathcal{F}_s\}_{s\ge 0}$-progressively measurable action process $a^{\boldsymbol{\pi}} = \{a^{\boldsymbol{\pi}}_s, t \le s \le T\}$ generated from $\boldsymbol{\pi}$, the corresponding state process $X^{\boldsymbol{\pi}} = \{X^{\boldsymbol{\pi}}_s, t \le s \le T\}$ follows

$$\mathrm{d}X^{\boldsymbol{\pi}}_s = b(s, X^{\boldsymbol{\pi}}_s, a^{\boldsymbol{\pi}}_s)\mathrm{d}s + \sigma(s, X^{\boldsymbol{\pi}}_s, a^{\boldsymbol{\pi}}_s)\mathrm{d}W_s,\ \ s \in [t, T];\ \ X^{\boldsymbol{\pi}}_t = x \tag{3}$$

defined on $(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_s\}_{s\ge 0})$. Moreover, following Wang et al. (2020), we add a regularizer to the reward function to encourage exploration (represented by the stochastic policy), leading to

$$\begin{aligned} J(t, x; \boldsymbol{\pi}) =& \mathbb{E}^{\mathbb{P}}\Big[\int_t^T e^{-\beta(s-t)} \left[r(s, X^{\boldsymbol{\pi}}_s, a^{\boldsymbol{\pi}}_s) + \gamma p\big(s, X^{\boldsymbol{\pi}}_s, a^{\boldsymbol{\pi}}_s, \boldsymbol{\pi}(\cdot|s, X^{\boldsymbol{\pi}}_s)\big)\right] \mathrm{d}s \\ & + e^{-\beta(T-t)} h(X^{\boldsymbol{\pi}}_T) \Big| X^{\boldsymbol{\pi}}_t = x \Big], \end{aligned} \tag{4}$$

1. Note that a single uniform random variable Z can produce many *independent* random variables having density functions. No dynamics are needed for these random variables and they are all independent of each other and of the Brownian motion. The independence means it makes no difference if these variables are given all at once at time 0 or are revealed as time evolves. We opt for the (mathematically speaking) easier construction where these are all defined using one single uniform Z. Meanwhile, $\mathbb{P}$ is the product extension from $\mathbb{P}^W$; the two probability measures coincide when restricted to $\mathcal{F}^W_T$.

2. Here we assume that the action space $\mathcal{A}$ is continuous and randomization is restricted to those distributions that have density functions. The analysis and results of this paper can be easily extended to the cases of discrete action spaces and/or randomization with probability mass functions.

where $\mathbb{E}^{\mathbb{P}}$ is the expectation w.r.t. both the Brownian motion and the action randomization. In the above, $p : [0,T] \times \mathbb{R}^d \times \mathcal{A} \times \mathcal{P}(\mathcal{A}) \mapsto \mathbb{R}$ is the regularizer and $\gamma \geq 0$ a weighting parameter on exploration, also known as the *temperature* parameter. Wang et al. (2020) take the differential entropy as the regularizer, which corresponds to

$$p\big(t, x, a, \pi(\cdot)\big) = \int_{\mathcal{A}} -\pi(z) \log \pi(z) \mathrm{d}z,$$

or equivalently,

$$p\big(t, x, a, \pi(\cdot)\big) = -\log \pi(a).$$

Through a law of large number argument, Wang et al. (2020) show that $\{X_s^{\boldsymbol{\pi}}, t \leq s \leq T\}$ has the same distribution as the solution to the following SDE, denoted by $\{\tilde{X}_s^{\boldsymbol{\pi}}, t \leq s \leq T\}$:

$$\mathrm{d}X_s = \tilde{b}\big(s, X_s, \boldsymbol{\pi}(\cdot|s, X_s)\big)\mathrm{d}t + \tilde{\sigma}\big(s, X_s, \boldsymbol{\pi}(\cdot|s, X_s)\big)\mathrm{d}W_s,\ s \in [t,T]; \quad X_t = x, \tag{5}$$

where

$$\tilde{b}\big(s, x, \pi(\cdot)\big) = \int_{\mathcal{A}} b(s, x, a)\pi(a)\mathrm{d}a,\ \tilde{\sigma}\big(s, x, \pi(\cdot)\big) = \sqrt{\int_{\mathcal{A}} \sigma^2(s, x, a)\pi(a)\mathrm{d}a}.$$

Moreover, the reward function (4) is identical to

$$\begin{aligned} J(t, x; \boldsymbol{\pi}) =& \mathbb{E}^{\mathbb{P}^W}\Bigg[\int_t^T e^{-\beta(s-t)} \int_{\mathcal{A}} [r(s, \tilde{X}_s^{\boldsymbol{\pi}}, a) + \gamma p\big(s, \tilde{X}_s^{\boldsymbol{\pi}}, a, \boldsymbol{\pi}(\cdot|s, \tilde{X}_s^{\boldsymbol{\pi}})\big)]\boldsymbol{\pi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}s \\ &+ e^{-\beta(T-t)} h(\tilde{X}_T^{\boldsymbol{\pi}})\Big|\tilde{X}_t^{\boldsymbol{\pi}} = x\Bigg]. \end{aligned} \tag{6}$$

Mathematically, (5) and (6) together form a so-called *relaxed stochastic control* problem where the effect of individually sampled actions has been averaged out (over the randomization/exploration) and, hence, one can focus on how a policy $\boldsymbol{\pi}$ impacts the *distribution* of the "averaged" state $\tilde{X}$; see Wang et al. (2020).

Here, $J(t, x; \boldsymbol{\pi})$ is called the *value function* of the policy $\boldsymbol{\pi}$, and the task of RL is to find

$$J^*(t, x) = \max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} J(t, x; \boldsymbol{\pi}), \tag{7}$$

where $\boldsymbol{\Pi}$ stands for the set of admissible policies. The following gives the precise definition of admissible (feedback) policies.

Definition 1 *A policy $\boldsymbol{\pi} = \boldsymbol{\pi}(\cdot|\cdot, \cdot)$ is called admissible if*

(i) $\boldsymbol{\pi}(\cdot|t, x) \in \mathcal{P}(\mathcal{A})$, and $\boldsymbol{\pi}(a|t, x) : (t, x, a) \in [0,T] \times \mathbb{R}^d \times \mathcal{A} \mapsto \mathbb{R}$ is measurable;

(ii) the SDE (5) admits a unique weak solution (in the sense of distribution) for any initial $(t, x) \in [0,T] \times \mathbb{R}^d$;

(iii) $\int_{\mathcal{A}} |r(t, x, a) + \gamma p\big(t, x, a, \boldsymbol{\pi}(\cdot|t, x)\big)|\boldsymbol{\pi}(a|t, x)\mathrm{d}a \leq C(1 + |x|^\mu)$, $\forall (t, x)$ where $C > 0$ and $\mu \geq 1$ are constants;

(iv) $\boldsymbol{\pi}(a|t,x)$ *is continuous in* (t,x) *and uniformly Lipschitz continuous in* x *in the total variation distance, i.e., for each fixed* a, $\int_{\mathcal{A}} |\boldsymbol{\pi}(a|t,x) - \boldsymbol{\pi}(a|t',x')|\mathrm{d}a \to 0$ *as* $(t',x') \to (t,x)$*, and there is a constant* $C > 0$ *independent of* (t,a) *such that*

$$\int_{\mathcal{A}} |\boldsymbol{\pi}(a|t,x) - \boldsymbol{\pi}(a|t,x')|\mathrm{d}a \le C|x - x'|, \;\; \forall x, x' \in \mathbb{R}^d.$$

The conditions required in the above definition, while not necessarily the weakest ones, are to theoretically guarantee the well-posedness of the control problem (5)–(6). This is implied by the following result.

Lemma 2 *Let Assumptions 1 hold and* $\boldsymbol{\pi}$ *be a given admissible policy. Then the SDE* (5) *admits a unique strong solution. Moreover, for any* $\mu \ge 2$*, the solution satisfies the growth condition* $\mathbb{E}^{\mathbb{P}^W}\left[\max_{t\le s\le T} |\tilde{X}_s^{\boldsymbol{\pi}}|^{\mu} \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x\right] \le C(1 + |x|^{\mu})$ *for some constant* $C = C(\mu)$*. Finally, the expected payoff* (6) *is finite.*

We stress that the solution to (5), $\{\tilde{X}_s^{\boldsymbol{\pi}}, t \le s \le T\}$, is the *average* of the sample trajectories over infinitely many randomized actions and is in itself *not* a sample trajectory *nor* observable. The stochastic relaxed control problem (5)– (6), introduced in Wang et al. (2020), just provides a framework for *theoretical* analysis. In contrast, the solution to (3), $\{X_s^{\boldsymbol{\pi}}, t \le s \le T\}$, is a sample trajectory under a realization of action sequence, $\{a_s^{\boldsymbol{\pi}}, t \le s \le T\}$, generated from the policy $\boldsymbol{\pi}$, and can indeed be observed. Meanwhile, the difference between (3) and (1) is that actions in the former are randomized: $a^{\boldsymbol{\pi}}$ is also driven by the randomization and hence is *not* $\mathcal{F}_t^W$-adapted. By taking the expectation w.r.t. the action randomization, the expectation in (4) reduces to the expectation in (6). In other words, the problem (5)–(6) is mathematically equivalent to the problem (3)–(4); yet they serve different purposes in our study: the former provides a framework for theoretical analysis of the value function while the latter directly involves observable samples.

Unlike most RL problems that are formulated in an infinite planning horizon (known as *continuing tasks*), the current paper mainly focuses on a finite horizon setting (known as *episodic tasks*). Finite horizons reflect limited lifespans of real-life tasks, e.g., a trader sells a financial contract with a maturity date, a robot finishes a task before a deadline, and a game player strives to pass a checkpoint given a time limit. If we let $T \to \infty$, under suitable regularity conditions (e.g., when β is large enough) our formulation covers the discounted formulation of the continuing tasks. In addition, later we will consider an ergodic setting as an alternative formulation for continuing tasks.

3. Theoretical Foundation of Actor–Critic Algorithms

An actor-critic (AC) algorithm consists of two parts: to estimate the value function of a given policy and to update (improve) the policy. In this section, we provide the theoretical analysis to guide devising such an algorithm through policy evaluation (PE) and policy gradient (PG).

3.1 Policy Evaluation

Jia and Zhou (2021) take a martingale perspective to characterize PE as well as its link to numerically solving a linear partial differential equation (PDE). However, they consider only *deterministic* policies (i.e. no randomization/exploration), without explicitly involving actions sampled from a stochastic policy. The extension to the case of stochastic policies is non-trivial and specific statements of the corresponding results are important for the subsequent PG and AC algorithm design; so we present and prove them here.

For a given stochastic policy $\boldsymbol{\pi}$, $J(\cdot,\cdot;\boldsymbol{\pi})$ can be characterized by a PDE based on the celebrated Feynman–Kac formula (cf. Karatzas and Shreve, 2014) which also holds true for the relaxed control setting.

Lemma 3 *Assume there is a unique viscosity solution $v \in C\big([0,T]\times\mathbb{R}^d\big)$ to the following PDE:*

$$\int_{\mathcal{A}}\big[\mathcal{L}^a v(t,x)+r\big(t,x,a\big)+\gamma p\big(t,x,a,\boldsymbol{\pi}(\cdot|t,x)\big)-\beta v(t,x)\big]\boldsymbol{\pi}(a|t,x)\mathrm{d}a=0,\ (t,x)\in[0,T)\times\mathbb{R}^d, \tag{8}$$

with the terminal condition $v(T,x)=h(x)$, $x\in\mathbb{R}^d$, which satisfies $|v(t,x)|\le C(1+|x|^\mu)$ for a constant $C>0$ and $\mu\ge 1$. Then v is the value function, that is, $v(t,x)=J(t,x;\boldsymbol{\pi})$ for all $(t,x)\in[0,T)\times\mathbb{R}^d$.

To avoid unduly technicalities, we assume throughout this paper that the value function $J\in C^{1,2}\big([0,T)\times\mathbb{R}^d\big)\cap C\big([0,T]\times\mathbb{R}^d\big)$. There is a rich literature on conditions ensuring the unique existence and regularity of the viscosity solution to the type of equations like (8); but see Tang et al. (2021) for some latest results.

The following is the main theoretical result underpinning PE, extended from the setting of deterministic feedback policies in Jia and Zhou (2021) to that of stochastic policies.

Theorem 4 *A function $J(\cdot,\cdot;\boldsymbol{\pi})$ is the value function associated with the policy $\boldsymbol{\pi}$ if and only if it satisfies terminal condition $J(T,x;\boldsymbol{\pi})=h(x)$, and for any initial $(t,x)\in[0,T)\times\mathbb{R}^d$:*

$$e^{-\beta s}J(s,\tilde{X}^{\boldsymbol{\pi}}_s;\boldsymbol{\pi})+\int_t^s e^{-\beta s'}\int_{\mathcal{A}}[r(s',\tilde{X}^{\boldsymbol{\pi}}_{s'},a)+\gamma p\big(s',\tilde{X}^{\boldsymbol{\pi}}_{s'},a,\boldsymbol{\pi}(\cdot|s',\tilde{X}^{\boldsymbol{\pi}}_{s'})\big)]\boldsymbol{\pi}(a|s',\tilde{X}^{\boldsymbol{\pi}}_{s'})\mathrm{d}a\mathrm{d}s'$$

is an $(\mathcal{F}^{\tilde{X}^{\boldsymbol{\pi}}},\mathbb{P}^W)$-martingale on $[t,T]$. Moreover, it is also equivalent to the martingale orthogonality condition:

$$\mathbb{E}^{\mathbb{P}}\int_0^T \xi_t\Big[\mathrm{d}J(t,X^{\boldsymbol{\pi}}_t;\boldsymbol{\pi})+r(t,X^{\boldsymbol{\pi}}_t,a^{\boldsymbol{\pi}}_t)\mathrm{d}t+\gamma p\big(t,X^{\boldsymbol{\pi}}_t,a^{\boldsymbol{\pi}}_t,\boldsymbol{\pi}(\cdot|t,X^{\boldsymbol{\pi}}_t)\big)\mathrm{d}t-\beta J(t,X^{\boldsymbol{\pi}}_t;\boldsymbol{\pi})\mathrm{d}t\Big]=0, \tag{9}$$

for any $\xi\in L^2_{\mathcal{F}^{X^{\boldsymbol{\pi}}}}\big([0,T];J(\cdot,X^{\boldsymbol{\pi}}_\cdot;\boldsymbol{\pi})\big)$.

In the above theorem, ξ is called a *test function* by convention, although in general it is actually a stochastic process.

In RL, one typically employs function approximation for learning functions of interest. Specifically, for PE, one uses a family of parameterized functions $J^\theta\equiv J^\theta(\cdot,\cdot;\boldsymbol{\pi})$ on $[0,T]\times$

$\mathbb{R}^d$ to approximate J, where $\theta \in \Theta \subseteq \mathbb{R}^{L_\theta}$, and the problem is reduced to finding the "best" (in some sense) θ. We make the following assumption on these function approximators to be used. (Henceforth we may drop $\boldsymbol{\pi}$ from $J^\theta(\cdot,\cdot;\boldsymbol{\pi})$ whenever no ambiguity arises.)

Assumption 2 *For all $\theta \in \Theta$, $J^\theta \in C^{1,2}\big([0,T)\times\mathbb{R}^d\big)\cap C\big([0,T]\times\mathbb{R}^d\big)$ and satisfies the polynomial growth condition in x. Moreover, $J^\theta(t,x)$ is a smooth function in θ with $\frac{\partial J^\theta}{\partial\theta}, \frac{\partial^2 J^\theta}{\partial\theta^2} \in C^{1,2}\big([0,T)\times\mathbb{R}^d\big)\cap C\big([0,T]\times\mathbb{R}^d\big)$ satisfying the polynomial growth condition in x.*

Thanks to the martingale characterization in Theorem 4, the PE algorithms developed in Jia and Zhou (2021) can be adapted to the current setting in a straightforward manner. We now summarize them.

(i) Minimize the martingale loss function (offline):

$$\min_{\theta\in\Theta}\mathbb{E}^{\mathbb{P}}\Bigg[\int_0^T\Bigg(e^{-\beta T}h(X_T^{\boldsymbol{\pi}})-e^{-\beta t}J^\theta(t,X_t^{\boldsymbol{\pi}}) + \int_t^T e^{-\beta s}[r(s,X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}})+\gamma p\big(s,X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}},\boldsymbol{\pi}(\cdot|s,X_s^{\boldsymbol{\pi}})\big)]\mathrm{d}s\Bigg)^2\mathrm{d}t\Bigg].$$

This objective corresponds to the gradient Monte-Carlo algorithm for discrete MDPs (Sutton and Barto, 2018).

(ii) Solve the martingale orthogonality condition (online/offline):

$$\mathbb{E}^{\mathbb{P}}\Bigg\{\int_0^T\xi_t\big[\mathrm{d}J^\theta(t,X_t^{\boldsymbol{\pi}})+r(t,X_t^{\boldsymbol{\pi}},a_t^{\boldsymbol{\pi}})\mathrm{d}t+\gamma p\big(t,X_t^{\boldsymbol{\pi}},a_t^{\boldsymbol{\pi}},\boldsymbol{\pi}(\cdot|t,X_t^{\boldsymbol{\pi}})\big)\mathrm{d}t-\beta J^\theta(t,X_t^{\boldsymbol{\pi}})\mathrm{d}t\big]\Bigg\}=0.$$

This objective corresponds to various (semi-gradient) TD algorithms and their variants for MDPs (Sutton, 1988; Bradtke and Barto, 1996), depending on the choices of the test function ξ.

(iii) Minimize a quadratic form of the martingale orthogonality condition (online/offline):

$$\min_{\theta\in\Theta}\mathbb{E}^{\mathbb{P}}\Bigg\{\int_0^T\xi_t\big[\mathrm{d}J^\theta(t,X_t^{\boldsymbol{\pi}})+r(t,X_t^{\boldsymbol{\pi}},a_t^{\boldsymbol{\pi}})\mathrm{d}t+\gamma p\big(t,X_t^{\boldsymbol{\pi}},a_t^{\boldsymbol{\pi}},\boldsymbol{\pi}(\cdot|t,X_t^{\boldsymbol{\pi}})\big)\mathrm{d}t-\beta J^\theta(t,X_t^{\boldsymbol{\pi}})\mathrm{d}t\big]\Bigg\}^\top A\mathbb{E}^{\mathbb{P}}\Bigg\{\int_0^T\xi_t\big[\mathrm{d}J^\theta(t,X_t^{\boldsymbol{\pi}})+r(t,X_t^{\boldsymbol{\pi}},a_t^{\boldsymbol{\pi}})\mathrm{d}t+\gamma p\big(t,X_t^{\boldsymbol{\pi}},a_t^{\boldsymbol{\pi}},\boldsymbol{\pi}(\cdot|t,X_t^{\boldsymbol{\pi}})\big)\mathrm{d}t-\beta J^\theta(t,X_t^{\boldsymbol{\pi}})\mathrm{d}t\big]\Bigg\},$$

where A is a positive definite matrix of a suitable size. Typical choices are $A=I$ or $A=\big(\mathbb{E}^{\mathbb{P}}[\int_0^T\xi_t\xi_t^\top\mathrm{d}t]\big)^{-1}$. This objective corresponds to the gradient TD algorithms and their variants for MDPs (Sutton et al., 2008, 2009; Maei et al., 2009).

In the above, the choice of the parametric family J^θ may be guided by exploiting some special structure of the underlying problem; see Wang and Zhou (2020) for an example. More general choices include linear combinations of some basis functions or neural networks. On the other hand, common choices of the test functions are $\xi_t=\frac{\partial J^\theta}{\partial\theta}(t,X_t^{\boldsymbol{\pi}})$ or $\xi_t=\int_0^t\lambda^{s-t}\frac{\partial J^\theta}{\partial\theta}(s,X_s^{\boldsymbol{\pi}})\mathrm{d}s$. Refer to the aforementioned references for details, and in particular to Jia and Zhou (2021) for the continuous setting. Finally, when implementing these algorithms we need to discretize time, and the convergence when the mesh size goes to zero is established in Jia and Zhou (2021), which can be readily extended to the current setting.

3.2 Policy Gradient

Given an admissible policy, suppose we have carried out the PE step and obtained an estimate of the corresponding value function. The next step is PG, namely, to estimate the gradient of the (learned) value function w.r.t. the policy. Specifically, let $\boldsymbol{\pi}^\phi$ be a parametric family of policies with the parameter $\phi \in \Phi \subset \mathbb{R}^{L_\phi}$. We aim to compute the policy gradient $g(t,x;\phi) := \frac{\partial}{\partial \phi} J(t,x;\boldsymbol{\pi}^\phi) \in \mathbb{R}^{L_\phi}$ at the current time–state pair (t,x). Here and throughout we always assume $\boldsymbol{\pi}^\phi$ is an admissible policy.

Based on the PDE characterization (8) of the value function, we take derivative in ϕ on both sides of (8), with $v(t,x)$ replaced by $J(t,x;\boldsymbol{\pi}^\phi)$, to get a new PDE satisfied by $g(t,x;\phi)$:

$$
\begin{cases}
\displaystyle\int_{\mathcal{A}} \Big\{ \big[\mathcal{L}^a g(t,x;\phi) - \beta g(t,x;\phi) + \gamma q(t,x,a,\phi)\big] \boldsymbol{\pi}^\phi(a|t,x) \\
\qquad + \big[\mathcal{L}^a J(t,x;\boldsymbol{\pi}^\phi) + r(t,x,a) + \gamma p\big(t,x,a,\boldsymbol{\pi}^\phi(\cdot|t,x)\big) - \beta J(t,x;\boldsymbol{\pi}^\phi)\big] \dfrac{\partial \boldsymbol{\pi}^\phi}{\partial \phi}(a|t,x) \Big\} \mathrm{d}a = 0, \\
g(T,x;\phi) = 0,
\end{cases}
\tag{10}
$$

where $q(t,x,a,\phi) = \frac{\partial}{\partial \phi} p\big(t,x,a,\boldsymbol{\pi}^\phi(\cdot|t,x)\big)$ that maps $[0,T] \times \mathbb{R}^d \times \mathcal{A} \times \Phi$ to $\mathbb{R}^{L_\phi}$. Note (10) is a *system* of L_ϕ equations, and $\mathcal{L}^a g$ denotes applying the operator $\mathcal{L}^a$ to each component of the $\mathbb{R}^{L_\phi}$-valued function $g(t,\cdot;\phi)$.

Define

$$
\begin{aligned}
\check{r}(t,x,a;\phi) =& \Big[\mathcal{L}^a J(t,x;\boldsymbol{\pi}^\phi) + r(t,x,a) + \gamma p\big(t,x,a,\boldsymbol{\pi}^\phi(\cdot|t,x)\big) - \beta J(t,x;\boldsymbol{\pi}^\phi)\Big] \frac{\frac{\partial \boldsymbol{\pi}^\phi}{\partial \phi}(a|t,x)}{\boldsymbol{\pi}^\phi(a|t,x)} \\
& + \gamma q(t,x,a,\phi) \\
=& \Big[\mathcal{L}^a J(t,x;\boldsymbol{\pi}^\phi) + r(t,x,a) + \gamma p\big(t,x,a,\boldsymbol{\pi}^\phi(\cdot|t,x)\big) - \beta J(t,x;\boldsymbol{\pi}^\phi)\Big] \frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^\phi(a|t,x) \\
& + \gamma q(t,x,a,\phi),
\end{aligned}
$$

which is again a function that maps $[0,T] \times \mathbb{R}^d \times \mathcal{A} \times \Phi$ to $\mathbb{R}^{L_\phi}$. Then (10) can be written as

$$
\int_{\mathcal{A}} \big[\mathcal{L}^a g(t,x;\phi) - \beta g(t,x;\phi) + \check{r}(t,x,a;\phi)\big] \boldsymbol{\pi}^\phi(a|t,x) \mathrm{d}a = 0, \;\; g(T,x;\phi) = 0. \tag{11}
$$

Observe that (11) has the similar form to (8). Thus a Feynman–Kac formula (similar to Lemma 3) represents g as

$$
\begin{aligned}
g(t,x;\phi) =& \mathbb{E}^{\mathbb{P}} \left[\int_t^T e^{-\beta(s-t)} \check{r}(s, X_s^{\boldsymbol{\pi}^\phi}, a_s^{\boldsymbol{\pi}^\phi}; \phi) \mathrm{d}s \Big| X_t^{\boldsymbol{\pi}^\phi} = x \right] \\
=& \mathbb{E}^{\mathbb{P}^W} \left[\int_t^T e^{-\beta(s-t)} \int_{\mathcal{A}} \check{r}(s, \tilde{X}_s^{\boldsymbol{\pi}^\phi}, a; \phi) \boldsymbol{\pi}^\phi(a|s, \tilde{X}_s^{\boldsymbol{\pi}^\phi}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}^\phi} = x \right].
\end{aligned}
\tag{12}
$$

Therefore, computing PG boils down mathematically to a PE problem with a *different* reward function. Indeed, the task here is much easier because we only need to compute the function *value*, $g(t,x;\phi)$, via (12) at *some* (t,x) along a sample trajectory, instead of learning the *entire* function $g(\cdot,\cdot;\phi)$ as in PE. However, unlike a normal PE problem, the new reward function $\check{r}$ involves the operator $\mathcal{L}^a$ applied to J which can not be observed nor computed without the knowledge of the environment.

The remedy to overcome this difficulty rests with Itô's lemma and martingality. We now provide an informal argument for explanation before presenting the formal result. Suppose at time t, an action a is generated from $\boldsymbol{\pi}^\phi(\cdot|t,X_t)$ and applied to the system within a small time window $[t,t+\Delta t]$. Apply Itô's lemma to obtain

$$J(t+\Delta t, X^a_{t+\Delta t};\boldsymbol{\pi}^\phi) - J(t,X_t;\boldsymbol{\pi}^\phi) = \int_t^{t+\Delta t} \mathcal{L}^a J(s,X^a_s;\boldsymbol{\pi}^\phi)\mathrm{d}s + \frac{\partial J}{\partial x}(s,X^a_s;\boldsymbol{\pi}^\phi)^\top \sigma_s \mathrm{d}W_s.$$

Therefore,

$$\begin{aligned}
&\check{r}(s,X^a_s,a;\phi)\mathrm{d}s \\
\equiv& \Big[\mathcal{L}^a J(s,X^a_s;\boldsymbol{\pi}^\phi) + r(s,X^a_s,a) + \gamma p\big(s,X^a_s,a,\boldsymbol{\pi}^\phi(\cdot|s,X^a_s)\big) - \beta J(s,X^a_s;\boldsymbol{\pi}^\phi)\Big]\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a|s,X^a_s)\mathrm{d}s \\
&+\gamma q(s,X^a_s,a,\phi)\mathrm{d}s \\
\approx& \frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a|s,X^a_s)\Big\{\mathrm{d}J(s,X^a_s;\boldsymbol{\pi}^\phi) + \big[r(s,X^a_s,a) + \gamma p\big(s,X^a_s,a,\boldsymbol{\pi}^\phi(\cdot|s,X^a_s)\big) - \beta J(s,X^a_s;\boldsymbol{\pi}^\phi)\big]\mathrm{d}s \\
&-\frac{\partial J}{\partial x}(s,X^a_s;\boldsymbol{\pi})^\top\sigma_s\mathrm{d}W_s\Big\} + \gamma q(s,X^a_s,a,\phi)\mathrm{d}s.
\end{aligned} \tag{13}$$

Since the stochastic integral w.r.t. the $\mathrm{d}W$ term above is a martingale (under suitable regularity conditions), such a term, even if unknown, does not contribute to the expectation and thus can be ignored. As a result, $\check{r}$ can be incrementally estimated based on observations of samples and the learned value function.

Before stating the main result of this paper, we impose the following technical conditions on the policy approximators.

Assumption 3 *$\boldsymbol{\pi}^\phi(a|t,x)$ is smooth in $\phi\in\Phi$ for all (t,x,a). Moreover, $\int_{\mathcal{A}}|\check{r}(t,x,a;\phi)|\boldsymbol{\pi}^\phi(a|t,x)\mathrm{d}a \le C(1+|x|^\mu)$ for all (t,x,ϕ), where $C>0,\mu\ge 1$ are constants. Furthermore, $\int_{\mathcal{A}}|\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a|t,x)|^2\boldsymbol{\pi}^\phi(a|t,x)\mathrm{d}a$ is continuous in (t,x) for all $\phi\in\Phi$.*

Theorem 5 *Given an admissible parameterized policy $\boldsymbol{\pi}^\phi$, its policy gradient $g(t,x;\phi) = \frac{\partial}{\partial\phi}J(t,x;\boldsymbol{\pi}^\phi)$ admits the following representation:*

$$\begin{aligned}
g(t,x;\phi) =& \mathbb{E}^{\mathbb{P}}\Bigg[\int_t^T e^{-\beta(s-t)}\Big\{\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a^{\boldsymbol{\pi}^\phi}_s|s,X^{\boldsymbol{\pi}^\phi}_s)\Big(\mathrm{d}J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi) \\
&+[r(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s) + \gamma p\big(s,X^{\boldsymbol{\pi}}_s,a^{\boldsymbol{\pi}^\phi}_s,\boldsymbol{\pi}^\phi(\cdot|s,X^{\boldsymbol{\pi}^\phi}_s)\big) - \beta J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)]\mathrm{d}s\Big) \\
&+\gamma q(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\phi)\mathrm{d}s\Big\}\Big|X^{\boldsymbol{\pi}^\phi}_t = x\Bigg],\quad (t,x)\in[0,T]\times\mathbb{R}^d.
\end{aligned} \tag{14}$$

Once again, all the terms inside the expectation above are all computable given samples (including action trajectories and the corresponding state trajectories) on $[t,T]$, together with an estimated value function J (obtained in the previous PE step). Note that the expectation (14) gives the gradient of the value function w.r.t. any policy, which is not 0 in general.

Observing (14) more closely, we can write $g(t,x;\phi)=g_1(t,x;\phi)+g_2(t,x;\phi)$ where

$$\begin{aligned}g_1(t,x;\phi)= \quad & \mathbb{E}^{\mathbb{P}}\Bigg[\textstyle\int_t^T e^{-\beta(s-t)}\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^{\phi}(a_s^{\boldsymbol{\pi}^\phi}|s,X_s^{\boldsymbol{\pi}^\phi})\Big(\mathrm{d}J(s,X_s^{\boldsymbol{\pi}^\phi};\boldsymbol{\pi}^\phi)\\ & +[r(s,X_s^{\boldsymbol{\pi}^\phi},a_s^{\boldsymbol{\pi}^\phi})+\gamma p\big(s,X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}^\phi},\boldsymbol{\pi}^\phi(\cdot|s,X_s^{\boldsymbol{\pi}^\phi})\big)-\beta J(s,X_s^{\boldsymbol{\pi}^\phi};\boldsymbol{\pi}^\phi)]\mathrm{d}s\Big)\Big|X_t^{\boldsymbol{\pi}^\phi}=x\Bigg]\end{aligned}$$

and

$$g_2(t,x;\phi)=\mathbb{E}^{\mathbb{P}}\left[\int_t^T e^{-\beta(s-t)}\gamma q(s,X_s^{\boldsymbol{\pi}^\phi},a_s^{\boldsymbol{\pi}^\phi},\phi)\mathrm{d}s\Big|X_t^{\boldsymbol{\pi}^\phi}=x\right].$$

The integrand in the expression of g_1 is the discounted derivative of the log-likelihood (log-pdf) that determines the direction, multiplied by a scalar term. This scalar term is actually the TD error in the continuous setting (Jia and Zhou, 2021) that also appears in the martingale orthogonality condition (9). Note that $g_1(t,x;\phi)\neq 0$ in general, because $\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a_s^{\boldsymbol{\pi}^\phi}|s,X_s^{\boldsymbol{\pi}^\phi})$ depends on the realization of $a_s^{\boldsymbol{\pi}^\phi}$, and hence is not $\mathcal{F}_s^{X^{\boldsymbol{\pi}^\phi}}$-measurable and does not qualify as a test function ξ in Theorem 4. On the other hand, g_2 comes entirely from the regularizer and vanishes should the latter be absent.

There are two equivalent forms of the representation (14), which can be used to add more flexibilities in designing PG algorithms and to optimize their performance. The first one is to add a "baseline" action-independent function $B(t,x)$ to the integrand in (14). Precisely, it follows from $a_s^{\boldsymbol{\pi}^\phi}\sim\boldsymbol{\pi}^\phi(\cdot|s,X_s^{\boldsymbol{\pi}^\phi})$ that

$$\mathbb{E}^{\mathbb{P}}\left[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a_s^{\boldsymbol{\pi}^\phi}|s,X_s^{\boldsymbol{\pi}^\phi})B(s,X_s^{\boldsymbol{\pi}^\phi})|X_s^{\boldsymbol{\pi}^\phi}\right]=0.$$

Hence, an alternative representation of (14) is

$$\begin{aligned}g(t,x;\phi)=&\mathbb{E}^{\mathbb{P}}\Bigg[\int_t^T e^{-\beta(s-t)}\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a_s^{\boldsymbol{\pi}^\phi}|s,X_s^{\boldsymbol{\pi}^\phi})\big]\big[\mathrm{d}J(s,X_s^{\boldsymbol{\pi}^\phi};\boldsymbol{\pi}^\phi)\\ &+[r(s,X_s^{\boldsymbol{\pi}^\phi},a_s^{\boldsymbol{\pi}^\phi})+\gamma p\big(s,X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}^\phi},\boldsymbol{\pi}^\phi(\cdot|s,X_s^{\boldsymbol{\pi}^\phi})\big)-\beta J(s,X_s^{\boldsymbol{\pi}^\phi};\boldsymbol{\pi}^\phi)-B(s,X_s^{\boldsymbol{\pi}^\phi})]\mathrm{d}s\big]\\ &+\gamma q(s,X_s^{\boldsymbol{\pi}^\phi},a_s^{\boldsymbol{\pi}^\phi},\phi)\mathrm{d}s\Big\}\Big|X_t^{\boldsymbol{\pi}^\phi}=x\Bigg].\end{aligned} \tag{15}$$

Including such a baseline function in the representation of PG goes back at least to Williams (1992). Sutton et al. (2000b) and Zhao et al. (2011) find that adding an appropriate baseline function can reduce the variance of the learning process. In particular, a common choice of baseline function, though not theoretically optimal, is the current value function, which leads to the so-called advantage AC algorithms (Degris et al., 2012; Mnih et al., 2016). Interestingly, without including any exogenous baseline function, the PG algorithms out of

(14) are exactly the continuous-time versions of the advantage AC algorithms. As such, we do not add other baseline functions for designing our algorithms below.

The second alternative form of (14) is to add an admissible test function to the derivative of the log-likelihood, based on Theorem 4. Specifically, suppose $\zeta \in L^2_{\mathcal{F}^{X^{\boldsymbol{\pi}^\phi}}}\big([0,T]; J(\cdot, X^{\boldsymbol{\pi}^\phi}_\cdot; \boldsymbol{\pi}^\phi)\big)$ is an $\mathbb{R}^{L_\phi}$-valued process. Then the policy gradient can also be represented by

$$
\begin{aligned}
g(t,x;\phi) =& \mathbb{E}^{\mathbb{P}}\Bigg[\int_t^T e^{-\beta(s-t)}\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a^{\boldsymbol{\pi}^\phi}_s|s,X^{\boldsymbol{\pi}^\phi}_s)+\zeta_s\big]\Big(\mathrm{d}J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi) \\
&+[r(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s)+\gamma p\big(s,X^{\boldsymbol{\pi}}_s,a^{\boldsymbol{\pi}^\phi}_s,\boldsymbol{\pi}^\phi(\cdot|s,X^{\boldsymbol{\pi}^\phi}_s)\big)-\beta J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)]\mathrm{d}s\Big) \\
&+\gamma q(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\phi)\mathrm{d}s\Big\}\Big|X^{\boldsymbol{\pi}^\phi}_t=x\Bigg],\quad (t,x)\in[0,T]\times\mathbb{R}^d.
\end{aligned}
\tag{16}
$$

As discussed before, we do not use (14) to approximate the function $g(\cdot,\cdot;\phi)$. Rather, at any current time–state (t,x), (14) gives the gradient of $J(t,x;\phi)$ in ϕ so that we can update ϕ in the most promising direction (based on the gradient ascent algorithm) to improve the value of J. However, the right hand side of (14) involves only the *future* trajectories from t; so Theorem 5 works only for the offline setting.

To treat the online case, assume that ϕ^* is the optimal point of $J(t,x;\boldsymbol{\pi}^\phi)$ for any (t,x) and that the first-order condition holds (e.g. when ϕ^* is an interior point).[3] Then $g(t,x;\phi^*)=0$. It thus follows from (10) that

$$
\begin{aligned}
0 =& \int_{\mathcal{A}}\Big\{\big[\mathcal{L}^a J(t,x;\boldsymbol{\pi}^{\phi^*})+r(t,x,a)+\gamma p\big(t,x,a,\boldsymbol{\pi}^{\phi^*}(\cdot|t,x)\big)-\beta J(t,x;\boldsymbol{\pi}^{\phi^*})\big]\frac{\partial\boldsymbol{\pi}^{\phi^*}}{\partial\phi}(a|t,x) \\
&+\gamma q(t,x,a,\phi^*)\boldsymbol{\pi}^{\phi^*}(a|t,x)\Big\}\mathrm{d}a \\
=& \int_{\mathcal{A}}\check{r}(t,x,a;\phi^*)\boldsymbol{\pi}^{\phi^*}(a|t,x)\mathrm{d}a.
\end{aligned}
\tag{17}
$$

This is the same type of equation as (8) involved in the Feynman–Kac formula. In the same way as (8) leading to Theorem 4, we can prove the following conclusion.

Theorem 6 *If there exists an interior optimal point ϕ^* that maximizes $J(0,x;\boldsymbol{\pi}^\phi)$ for any $x\in\mathbb{R}^d$, then*

$$
\begin{aligned}
0 =& \mathbb{E}^{\mathbb{P}}\Bigg[\int_0^T \eta_s\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^{\phi^*}(a^{\boldsymbol{\pi}^{\phi^*}}_s|s,X^{\boldsymbol{\pi}^{\phi^*}}_s)+\zeta_s\big]\big[\mathrm{d}J(s,X^{\boldsymbol{\pi}}_s;\boldsymbol{\pi}^{\phi^*}) \\
&+[r(s,X^{\boldsymbol{\pi}^{\phi^*}}_s,a^{\boldsymbol{\pi}^{\phi^*}}_s)+\gamma p\big(s,X^{\boldsymbol{\pi}^{\phi^*}}_s,a^{\boldsymbol{\pi}^{\phi^*}}_s,\boldsymbol{\pi}^{\phi^*}(\cdot|s,X^{\boldsymbol{\pi}^{\phi^*}}_s)\big)-\beta J(s,X^{\boldsymbol{\pi}^{\phi^*}}_s;\boldsymbol{\pi}^{\phi^*})]\mathrm{d}s\big] \\
&+\gamma q(s,X^{\boldsymbol{\pi}^{\phi^*}}_s,a^{\boldsymbol{\pi}^{\phi^*}}_s,\phi^*)\mathrm{d}s\Big\}\Big|X^{\boldsymbol{\pi}^{\phi^*}}_0=x\Bigg]
\end{aligned}
\tag{18}
$$

for any $\eta,\zeta\in L^2_{\mathcal{F}^{X^{\boldsymbol{\pi}^{\phi^*}}}}\big([0,T];J(\cdot,X^{\boldsymbol{\pi}^{\phi^*}}_\cdot;\boldsymbol{\pi}^{\phi^*})\big)$.

3. A theoretically optimal policy $\boldsymbol{\pi}^*$ indeed maximizes $J(t,x;\boldsymbol{\pi})$ for *any* (t,x), based on the verification theorem; see Yong and Zhou (1999).

If we take $\eta_s = e^{-\beta s}$, then the right hand side of (18) coincides with $g(0, x, \phi^*)$. However, though only a necessary condition, (18) contains infinitely many equations with different test functions η. More importantly, besides the flexibility of choosing different sets of test functions, (18) provides a way to derive a system of equations based on only *past* observations and, hence, enables online learning. For example, by taking $\eta_s = 0$ on $[t, T]$, (18) involves sample trajectories up to only the present time t. Thus, learning the optimal policy either offline or online boils down to solving a system of equations (with suitably chosen test functions) via stochastic approximation to find ϕ^*.

In sum, Theorems 5 and 6 foreshadow two different types of algorithms which we will develop in the next subsection.

3.3 Actor–Critic Algorithms

We now design actor–critic (AC) algorithms by combining the PE and the PG steps. For the former, Jia and Zhou (2021) develop two methods, those of martingale loss function and martingale orthogonality conditions, to devise several online/offline PE algorithms for the continuous setting. As discussed in Subsection 3.1, one can adopt any of these algorithms that is suitable for the given learning context and computational resource to estimate the value function of any given policy. So here we focus on how to update policy based on our previous theoretical analysis on PG.

First, in the offline setting where full state trajectories under any given policy can be repeatedly sampled and observed, the gradient of the value function w.r.t. the policy is given by (14), which can be estimated using future samples from any *current* time–state (t, x). That is, $g(t, x; \phi)$ is the gradient direction that would maximally improve the total reward at (t, x).

For online learning, as explained earlier, (14) is no longer implementable. Instead of computing gradients, we turn to (18) for directly solving the optimal policy. Specifically, at any current time t, we choose $\eta_s = 0$ for $s \in [t, T]$ so that the integral in (18) only utilizes past observations up to t, and hence is computable. Therefore, in the online setting one applies stochastic approximation to solve the optimal condition (18) in order to search for the optimal policy ϕ^*.

Recall that $J^\theta \equiv J^\theta(\cdot, \cdot)$, where $J^\theta(t, x) \in \mathbb{R}$, is a family of scalar functions on $(t, x) \in [0, T] \times \mathbb{R}^d$ parameterized by $\theta \in \Theta \subseteq \mathbb{R}^{L_\theta}$, and $\boldsymbol{\pi}^\phi \equiv \boldsymbol{\pi}^\phi(\cdot|\cdot, \cdot)$, where $\boldsymbol{\pi}^\phi(\cdot|t, x) \in \mathcal{P}(\mathcal{A})$, is a family of pdf-valued policy functions on $(t, x) \in [0, T] \times \mathbb{R}^d$ parameterized by $\phi \in \Phi \subseteq \mathbb{R}^{L_\phi}$. The aim of an AC algorithm is to find the optimal (θ, ϕ) jointly, by updating the two parameters alternatingly. Note that, although our problem is continuous in time, the final algorithmic implementation requires discretizing time. For simplicity, we use equally spaced mesh grid $t_k = k\Delta t$, with $k = 0, \cdots, K = \lfloor \frac{T}{\Delta t} \rfloor$.

We now present the following pseudo codes in Algorithms 1 and 2. Algorithm 1 is for offline-episodic learning, where full trajectories are sampled and observed repeatedly during different episodes and (θ, ϕ) are updated after one whole episode. Algorithm 2 is for online incremental learning, where only the past sample trajectory is available and (θ, ϕ) are updated in real-time incrementally.

Note that Algorithms 1 and 2 presented here are just for illustrative purpose; there is ample flexibility to devise their variants depending on the specific problems concerned. In

Algorithm 1 Offline–Episodic Actor–Critic Algorithm

Inputs: initial state x_0, horizon T, time step Δt, number of episodes N, number of mesh grids K, initial learning rates $\alpha_\theta, \alpha_\phi$ and a learning rate schedule function $l(\cdot)$ (a function of the number of episodes), functional form of the value function $J^\theta(\cdot,\cdot)$, functional form of the policy $\boldsymbol{\pi}^\phi(\cdot|\cdot,\cdot)$, functional form of the regularizer $p\big(t,x,a,\pi(\cdot)\big)$, functional forms of the test functions $\boldsymbol{\xi}(t, x_{\cdot\wedge t})$, $\boldsymbol{\zeta}(t, x_{\cdot\wedge t})$, and temperature parameter γ.

Required program: an environment simulator $(x', r) = Environment_{\Delta t}(t, x, a)$ that takes current time-state pair (t, x) and action a as inputs and generates state x' at time $t + \Delta t$ and the instantaneous reward r at time t.

Learning procedure:

Initialize θ, ϕ.

for episode $j = 1$ **to** N **do**

Initialize $k = 0$. Observe the initial state x_0 and store $x_{t_k} \leftarrow x_0$.

while $k < K$ **do**

Compute and store the test function $\xi_{t_k} = \boldsymbol{\xi}(t_k, x_{t_0}, \cdots, x_{t_k})$, $\zeta_{t_k} = \boldsymbol{\zeta}(t_k, x_{t_0}, \cdots, x_{t_k})$.

Generate action $a_{t_k} \sim \boldsymbol{\pi}^\phi(\cdot|t_k, x_{t_k})$.

Apply a_{t_k} to the environment simulator $(x, r) = Environment_{\Delta t}(t_k, x_{t_k}, a_{t_k})$, and observe the output new state x and reward r. Store $x_{t_{k+1}} \leftarrow x$ and $r_{t_k} \leftarrow r$.

Update $k \leftarrow k + 1$.

end while

Compute

$$\Delta\theta = \sum_{i=0}^{K-1} \xi_{t_i}\big[J^\theta(t_{i+1}, x_{t_{i+1}}) - J^\theta(t_i, x_{t_i}) + r_{t_i}\Delta t + \gamma p\big(t_i, x_{t_i}, a_{t_i}, \boldsymbol{\pi}^\phi(\cdot|t_i, x_{t_i})\big)\Delta t - \beta J^\theta(t_i, x_{t_i})\Delta t\big],$$

$$\begin{aligned}\Delta\phi = \sum_{i=0}^{K-1} e^{-\beta t_i}\Big\{ &\big[\frac{\partial}{\partial\phi}\log \boldsymbol{\pi}^\phi(a_{t_i}|t_i, x_{t_i}) + \zeta_{t_i}\big]\big[J^\theta(t_{i+1}, x_{t_{i+1}}) - J^\theta(t_i, x_{t_i}) + r_{t_i}\Delta t \\ &+ \gamma p\big(t_i, x_{t_i}, a_{t_i}, \boldsymbol{\pi}^\phi(\cdot|t_i, x_{t_i})\big)\Delta t - \beta J^\theta(t_i, x_{t_i})\Delta t\big] + \gamma\frac{\partial p}{\partial\phi}\big(t_i, x_{t_i}, a_{t_i}, \boldsymbol{\pi}^\phi(\cdot|t_i, x_{t_i})\big)\Delta t\Big\}.\end{aligned}$$

Update θ (policy evaluation) by

$$\theta \leftarrow \theta + l(j)\alpha_\theta\Delta\theta.$$

Update ϕ (policy gradient) by

$$\phi \leftarrow \phi + l(j)\alpha_\phi\Delta\phi.$$

end for

particular, the choice of test functions dictates in which sense we approximate the value

Algorithm 2 Online-Incremental Actor–Critic Algorithm

Inputs: initial state x_0, horizon T, time step Δt, number of mesh grids K, initial learning rates $\alpha_\theta, \alpha_\phi$ and learning rate schedule function $l(\cdot)$ (a function of the number of episodes), functional form of the value function $J^\theta(\cdot,\cdot)$, functional form of the policy $\boldsymbol{\pi}^\phi(\cdot|\cdot,\cdot)$, functional form of the regularizer $p\big(t,x,a,\pi(\cdot)\big)$, functional forms of the test functions $\boldsymbol{\xi}(t,x_{\cdot\wedge t}), \boldsymbol{\eta}(t,x_{\cdot\wedge t})$, $\boldsymbol{\zeta}(t,x_{\cdot\wedge t})$, and temperature parameter γ.

Required program: an environment simulator $(x',r) = \mathit{Environment}_{\Delta t}(t,x,a)$ that takes current time-state pair (t,x) and action a as inputs and generates state x' at time $t+\Delta t$ and the instantaneous reward r at time t.

Learning procedure:

Initialize θ, ϕ.

for episode $j=1$ **to** ∞ **do**

Initialize $k=0$. Observe the initial state x_0 and store $x_{t_k} \leftarrow x_0$.

while $k<K$ **do**

Compute test function $\xi_{t_k} = \boldsymbol{\xi}(t_k, x_{t_0},\cdots,x_{t_k})$, $\eta_{t_k} = \boldsymbol{\eta}(t_k, x_{t_0},\cdots,x_{t_k})$, and $\zeta_{t_k} = \boldsymbol{\zeta}(t_k, x_{t_0},\cdots,x_{t_k})$.

Generate action $a_{t_k} \sim \boldsymbol{\pi}^\phi(\cdot|t_k, x_{t_k})$.

Apply a_{t_k} to the environment simulator $(x,r) = \mathit{Environment}_{\Delta t}(t_k, x_{t_k}, a_{t_k})$, and observe the output new state x and reward r. Store $x_{t_{k+1}} \leftarrow x$ and $r_{t_k} \leftarrow r$.

Compute

$$
\begin{aligned}
&\delta = J^\theta(t_{k+1}, x_{t_{k+1}}) - J^\theta(t_k, x_{t_k}) + r_{t_k}\Delta t + \gamma p\big(t_k, x_{t_k}, a_{t_k}, \boldsymbol{\pi}^\phi(\cdot|t_k, x_{t_k})\big)\Delta t - \beta J^\theta(t_k, x_{t_k})\Delta t,\\
&\Delta\theta = \xi_{t_k}\delta,\\
&\Delta\phi = \eta_{t_k}\Big\{\big[\frac{\partial}{\partial\phi}\log \boldsymbol{\pi}^\phi(a_{t_k}|t_k, x_{t_k}) + \zeta_{t_k}\big]\delta + \gamma\frac{\partial p}{\partial \phi}\big(t_k, x_{t_k}, a_{t_k}, \boldsymbol{\pi}^\phi(\cdot|t_k, x_{t_k})\big)\Delta t\Big\}.
\end{aligned}
$$

Update θ (policy evaluation) by

$$\theta \leftarrow \theta + l(j)\alpha_\theta \Delta\theta.$$

Update ϕ (policy gradient) by

$$\phi \leftarrow \phi + l(j)\alpha_\phi \Delta\phi.$$

Update $k \leftarrow k+1$

end while

end for

function and policy.[4] For example, if we take the test functions $\xi_t = \frac{\partial J^\theta}{\partial\theta}(t,X_t)$, and $\eta_t = e^{-\beta t}$, then we have essentially TD(0) AC algorithms. If we take $\xi_t = \int_0^t \lambda^{t-s}\frac{\partial J^\theta}{\partial\theta}(s,X_s)\mathrm{d}s$,

4. See Jia and Zhou (2021) for detailed discussions on this point for the PE part. Also, to save computational and memory cost of algorithms, we usually choose test functions that can be computed incrementally. For example, in a TD(λ) algorithm, $\xi_{t_k} = \int_0^{t_k}\lambda^{t_k - s}\frac{\partial J^\theta}{\partial \theta}(s,X_s)\mathrm{d}s \approx \lambda^{\Delta t}\xi_{t_{k-1}} + \frac{\partial J^\theta}{\partial\theta}(t_k, X_{t_k})\Delta t$, and $\zeta_{t_k} \approx \lambda^{\Delta t}\zeta_{t_{k-1}} + \frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a^{\boldsymbol{\pi}^\phi}_{t_{k-1}}|t_{k-1}, X_{t_{k-1}})\Delta t$, which can be calculated recursively.

$\zeta_t = \int_0^{t-\Delta t} \lambda^{t-s} \frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^{\phi}(a_s^{\boldsymbol{\pi}^{\phi}}|s, X_s)\mathrm{d}s$, then we end up with TD(λ) algorithms (Sutton and Barto, 2018). Moreover, in the PE part of the algorithms we can also use other methods (online or offline) as summarized in Subsection 3.1.

Finally, we reiterate that the main purpose of this paper is to provide a theoretical foundation to guide designing AC algorithms, instead of comparing which algorithm performs better. As such, we only present the TD-type algorithms for illustration, acknowledging that there are multiple ways to combine PE and the newly developed PG methods to design new learning algorithms.

4. Extension to Ergodic Tasks

In this section we extend our results and algorithms to ergodic (long-term average) tasks, which are also commonly studied in the RL literature. The ergodic objective is one possible formulation of continuing tasks, in which a learning algorithm is based on only one single trajectory.

Consider a regularized ergodic objective function

$$\begin{aligned}
&\liminf_{T\to\infty} \frac{1}{T}\mathbb{E}^{\mathbb{P}^W}\left[\int_t^T \int_{\mathcal{A}} [r(\tilde{X}_s^{\boldsymbol{\pi}}, a) + \gamma p\big(\tilde{X}_s^{\boldsymbol{\pi}}, a, \boldsymbol{\pi}(\cdot|\tilde{X}_s^{\boldsymbol{\pi}})\big)]\boldsymbol{\pi}(a|\tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}s\Big|\tilde{X}_t^{\boldsymbol{\pi}} = x\right]\\
=&\liminf_{T\to\infty} \frac{1}{T}\mathbb{E}^{\mathbb{P}}\left[\int_t^T [r(X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) + \gamma p\big(X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\cdot|X_s^{\boldsymbol{\pi}})\big)]\mathrm{d}s\Big|X_t^{\boldsymbol{\pi}} = x\right],
\end{aligned}$$

where p is the regularizer and $\gamma \geq 0$ is the temperature parameter. Note that now the running reward, the regularizer and the policy do not depended on time explicitly due to the stationary nature of ergodic tasks.

One way to study an ergodic task is to connect it to a discounted, infinite horizon problem:

$$\begin{aligned}
&\mathbb{E}^{\mathbb{P}^W}\left[\int_t^\infty e^{-\beta(s-t)} \int_{\mathcal{A}} [r(\tilde{X}_s^{\boldsymbol{\pi}}, a) + \gamma p\big(\tilde{X}_s^{\boldsymbol{\pi}}, a, \boldsymbol{\pi}(\cdot|\tilde{X}_s^{\boldsymbol{\pi}})\big)]\boldsymbol{\pi}(a|\tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}s\Big|\tilde{X}_t^{\boldsymbol{\pi}} = x\right]\\
=&\mathbb{E}^{\mathbb{P}}\left[\int_t^\infty e^{-\beta(s-t)}[r(X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) + \gamma p\big(X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\cdot|X_s^{a})\big)]\mathrm{d}s\Big|X_t^{\boldsymbol{\pi}} = x\right].
\end{aligned}$$

It has been shown that, under suitable conditions, the optimal value function of the discounted infinite horizon problem converges to the optimal ergodic reward as the discount factor $\beta \to 0$; see, e.g., Borkar and Ghosh (1988, 1990); Bensoussan and Frehse (1992).

Here, we opt for a direct treatment of ergodic problems. According to Sutton and Barto (2018, page 249), ergodic tasks are actually better behaved than continuing tasks with discounting. For a systematic account of classical ergodic control theory in continuous time, see Arapostathis et al. (2012) and the references therein.

We first present the ergodic version of the Feynman–Kac formula.

Lemma 7 *Let $\boldsymbol{\pi} = \boldsymbol{\pi}(\cdot|\cdot)$ be a given (time-invariant) policy. Suppose there is a function $J(\cdot;\boldsymbol{\pi}) \in C^2(\mathbb{R}^d)$ and a scalar $V(\boldsymbol{\pi}) \in \mathbb{R}$ satisfying*

$$\int_{\mathcal{A}} \left[\mathcal{L}^a J(x;\boldsymbol{\pi}) + r(x,a) + \gamma p\big(x,a,\boldsymbol{\pi}(\cdot|x)\big)\right]\boldsymbol{\pi}(a|x)\mathrm{d}a - V(\boldsymbol{\pi}) = 0, \quad x \in \mathbb{R}^d. \tag{19}$$

Then for any $t \geq 0$,

$$\begin{aligned} V(\boldsymbol{\pi}) = & \quad \liminf_{T\to\infty} \tfrac{1}{T}\mathbb{E}^{\mathbb{P}^W}\left[\int_t^T \int_{\mathcal{A}}[r(\tilde{X}_s^{\boldsymbol{\pi}},a) + \gamma p\big(\tilde{X}_s^{\boldsymbol{\pi}},a,\boldsymbol{\pi}(\cdot|\tilde{X}_s^{\boldsymbol{\pi}})\big)]\boldsymbol{\pi}(a|\tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}s \middle| \tilde{X}_t^{\boldsymbol{\pi}} = x\right] \\ = & \quad \liminf_{T\to\infty} \tfrac{1}{T}\mathbb{E}^{\mathbb{P}}\left[\int_t^T [r(X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}}) + \gamma p\big(X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}},\boldsymbol{\pi}(\cdot|X_s^{\boldsymbol{\pi}})\big)]\mathrm{d}s \middle| X_t^{\boldsymbol{\pi}} = x\right]. \end{aligned} \tag{20}$$

Moreover, $J(X_t^{\boldsymbol{\pi}};\boldsymbol{\pi}) + \int_0^t [r(X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}}) + \gamma p\big(X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}},\boldsymbol{\pi}(\cdot|X_s^{\boldsymbol{\pi}})\big) - V(\boldsymbol{\pi})]\mathrm{d}s$ *is an* $(\mathcal{F}^{X^{\boldsymbol{\pi}}},\mathbb{P})$*-martingale.*

We emphasize that the solution to (19) is a *pair* of (J,V), where J is a function of the state and $V \in \mathbb{R}$ is a scalar. The long term average of the payoff does not depend on the initial state x nor the initial time t due to the ergodicity, and hence remains a constant as (20) implies. The function J, on the other hand, only represents the first-order approximation of long-run average and is not unique. Indeed, for any constant c, $(J+c,V)$ is also a solution to (19). We refer to V as the "value". Lastly, since the value does not depend on the initial time, we will fix the latter as 0 in the following discussions and applications of ergodic tasks.

For a given policy $\boldsymbol{\pi}$, the PE problem is now to find a function $J(\cdot;\boldsymbol{\pi})$ and a value $V \in \mathbb{R}$, such that

$$J(X_t^{\boldsymbol{\pi}};\boldsymbol{\pi}) + \int_0^t [r(X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}}) + \gamma p\big(X_s^{\boldsymbol{\pi}},a_s^{\boldsymbol{\pi}},\boldsymbol{\pi}(\cdot|X_s^{\boldsymbol{\pi}})\big) - V(\boldsymbol{\pi})]\mathrm{d}s$$

is a martingale. Following Jia and Zhou (2021), we can then design online PE algorithms based on the following martingale orthogonality conditions:

$$\mathbb{E}^{\mathbb{P}} \int_0^T \xi_t \Big\{ \mathrm{d}J(X_t^{\boldsymbol{\pi}};\boldsymbol{\pi}) + \big[r(X_t^{\boldsymbol{\pi}},a_t^{\boldsymbol{\pi}}) + \gamma p\big(X_t^{\boldsymbol{\pi}},a_t^{\boldsymbol{\pi}},\boldsymbol{\pi}(\cdot|X_t^{\boldsymbol{\pi}})\big) - V\big]\mathrm{d}t \Big\} = 0, \tag{21}$$

for any $T > 0$, any initial state x, and any test function $\xi \in L^2_{\mathcal{F}^{X^{\boldsymbol{\pi}}}}\big([0,T]; J(X_\cdot^{\boldsymbol{\pi}};\boldsymbol{\pi})\big)$.

We now focus on PG. Suppose we parameterize the policy by $\boldsymbol{\pi}^\phi$, we aim to estimate $\frac{\partial V(\boldsymbol{\pi}^\phi)}{\partial \phi}$. Taking derivative in ϕ in (19), we obtain

$$\begin{aligned} \frac{\partial V(\boldsymbol{\pi}^\phi)}{\partial \phi} = & \int_{\mathcal{A}} \big[\mathcal{L}^a J(x;\boldsymbol{\pi}^\phi) + r(x,a) + \gamma p\big(x,a,\boldsymbol{\pi}^\phi(\cdot|x)\big)\big] \frac{\partial \boldsymbol{\pi}^\phi(a|x)}{\partial \phi}\mathrm{d}a + \int_{\mathcal{A}} \mathcal{L}^a \frac{\partial J(x;\boldsymbol{\pi}^\phi)}{\partial \phi}\boldsymbol{\pi}^\phi(a|x)\mathrm{d}a \\ & + \gamma \int_{\mathcal{A}} \frac{\partial p\big(x,a,\boldsymbol{\pi}^\phi(\cdot|x)\big)}{\partial \phi}\boldsymbol{\pi}^\phi(a|x)\mathrm{d}a. \end{aligned}$$

Denote $q(x,a,\phi) := \frac{\partial}{\partial \phi} p\big(x,a,\boldsymbol{\pi}^\phi(\cdot|x)\big)$, $\check{r}(x,a;\phi) := \big[\mathcal{L}^a J(x;\boldsymbol{\pi}^\phi) + r(x,a) + \gamma p\big(x,a,\boldsymbol{\pi}(\cdot|x)\big)\big] \frac{\frac{\partial \boldsymbol{\pi}^\phi(a|x)}{\partial \phi}}{\boldsymbol{\pi}^\phi(a|x)} + \gamma q(x,a,\phi)$, and $g(x;\phi) := \frac{\partial}{\partial \phi} J(x;\boldsymbol{\pi}^\phi)$. Then

$$\int_{\mathcal{A}} [\mathcal{L}^a g(x;\phi) + \check{r}(x,a;\phi)]\boldsymbol{\pi}^\phi(a|x)\mathrm{d}a - \frac{\partial V(\boldsymbol{\pi}^\phi)}{\partial \phi} = 0.$$

Therefore, analogous to the case of episodic tasks, $\frac{\partial V(\boldsymbol{\pi}^\phi)}{\partial \phi}$ is the value corresponding to the long-term average of a different running reward, according to the ergodic Feynman–Kac formula (Lemma 7); that is

$$
\begin{aligned}
\frac{\partial V(\boldsymbol{\pi}^\phi)}{\partial \phi} = & \lim_{T\to\infty} \frac{1}{T}\mathbb{E}^{\mathbb{P}}\Big[\int_0^T \check{r}(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}; \phi)\mathrm{d}t \Big| X_0^{\boldsymbol{\pi}^\phi} = x\Big] \\
= & \lim_{T\to\infty} \frac{1}{T}\mathbb{E}^{\mathbb{P}^W}\Big[\int_0^T \int_{\mathcal{A}} \check{r}(\tilde{X}_t^{\boldsymbol{\pi}^\phi}, a; \phi)\boldsymbol{\pi}^\phi(a|\tilde{X}_t^{\boldsymbol{\pi}^\phi})\mathrm{d}a\mathrm{d}t \Big| \tilde{X}_0^{\boldsymbol{\pi}^\phi} = x\Big] \\
= & \lim_{T\to\infty} \frac{1}{T}\mathbb{E}^{\mathbb{P}^W}\Big[\int_0^T \int_{\mathcal{A}} \big\{ \big[\mathcal{L}^a J(\tilde{X}_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi) + r(\tilde{X}_t^{\boldsymbol{\pi}^\phi}, a) + \gamma p\big(\tilde{X}_t^{\boldsymbol{\pi}^\phi}, a, \boldsymbol{\pi}^\phi(\cdot|\tilde{X}_t^{\boldsymbol{\pi}^\phi})\big)\big] \\
& \times \frac{\partial}{\partial \phi}\log \boldsymbol{\pi}^\phi(a|\tilde{X}_t^{\boldsymbol{\pi}^\phi}) + \gamma q(\tilde{X}_t^{\boldsymbol{\pi}^\phi}, a, \phi)\big\}\boldsymbol{\pi}^\phi(a|\tilde{X}_t^{\boldsymbol{\pi}^\phi})\mathrm{d}a\mathrm{d}t \Big| \tilde{X}_0^{\boldsymbol{\pi}^\phi} = x\Big] \\
= & \lim_{T\to\infty} \frac{1}{T}\mathbb{E}^{\mathbb{P}}\Big[\int_0^T \big\{ \frac{\partial}{\partial \phi}\log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|X_t^{\boldsymbol{\pi}^\phi})\big[\mathrm{d}J(X_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi) + r(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi})\mathrm{d}t \\
& + \gamma p\big(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \boldsymbol{\pi}^\phi(\cdot|X_t^{\boldsymbol{\pi}^\phi})\big)\mathrm{d}t\big] + \gamma q(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \phi)\mathrm{d}t\big\} \Big| X_0^{\boldsymbol{\pi}^\phi} = x\Big] \\
= & \lim_{T\to\infty} \frac{1}{T}\mathbb{E}^{\mathbb{P}}\Big[\int_0^T \big\{ \frac{\partial}{\partial \phi}\log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|X_t^{\boldsymbol{\pi}^\phi})\big[\mathrm{d}J(X_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi) + r(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi})\mathrm{d}t \\
& + \gamma p\big(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \boldsymbol{\pi}^\phi(\cdot|X_t^{\boldsymbol{\pi}^\phi})\big)\mathrm{d}t - V\mathrm{d}t\big] + \gamma q(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \phi)\mathrm{d}t\big\} \Big| X_0^{\boldsymbol{\pi}^\phi} = x\Big],
\end{aligned}
\tag{22}
$$

where the last equality is due to

$$
\begin{aligned}
& \mathbb{E}^{\mathbb{P}}\Big[\int_0^T V\frac{\partial}{\partial \phi}\log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|X_t^{\boldsymbol{\pi}^\phi})\mathrm{d}t \Big| X_0^{\boldsymbol{\pi}^\phi} = x\Big] \\
= & V\mathbb{E}^{\mathbb{P}^W}\Big[\int_0^T \mathrm{d}t \int_{\mathcal{A}} \boldsymbol{\pi}^\phi(a|\tilde{X}_t^{\boldsymbol{\pi}^\phi})\frac{\partial}{\partial \phi}\log \boldsymbol{\pi}^\phi(a|\tilde{X}_t^{\boldsymbol{\pi}^\phi})\mathrm{d}a \Big| \tilde{X}_0^{\boldsymbol{\pi}^\phi} = x\Big] \\
= & V\mathbb{E}^{\mathbb{P}^W}\Big[\int_0^T \mathrm{d}t\frac{\partial}{\partial \phi} \int_{\mathcal{A}} \boldsymbol{\pi}^\phi(a|\tilde{X}_t^{\boldsymbol{\pi}^\phi})\mathrm{d}a \Big| \tilde{X}_0^{\boldsymbol{\pi}^\phi} = x\Big] = 0.
\end{aligned}
$$

An ergodic task is a continuing task so we are naturally interested in online algorithms only. We can design two algorithms based on the analysis above. The first one follows directly from the representation (22), in which the policy gradient is the expectation of a long-run average and hence can be estimated online incrementally by

$$
\frac{\partial}{\partial \phi}\log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|X_t^{\boldsymbol{\pi}^\phi})\Big[\mathrm{d}J(X_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi) + [r(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}) + \gamma p\big(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \boldsymbol{\pi}^\phi(\cdot|X_t^{\boldsymbol{\pi}^\phi})\big) - V]\mathrm{d}t\Big] + \gamma q(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \phi)\mathrm{d}t
$$

since it will converge to its stationary distribution as $t \to \infty$. Moreover, due to the martingale orthogonality condition (21), we can also add a test function ζ as we did in (16). Consequently, the algorithm updates ϕ by gradient ascent:

$$
\begin{aligned}
\phi \leftarrow \phi + \alpha_\phi\Big\{ & \big[\frac{\partial}{\partial \phi}\log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|X_t^{\boldsymbol{\pi}^\phi}) + \zeta_t\big]\Big[\mathrm{d}J(X_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi) \\
& + [r(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}) + \gamma p\big(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \boldsymbol{\pi}^\phi(\cdot|X_t^a)\big) - V]\mathrm{d}t\Big] + \gamma q(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \phi)\mathrm{d}t\Big\}.
\end{aligned}
$$

The second algorithm applies a test function η and stochastic approximation to solve the optimality condition as in Theorem 6, by updating

$$\phi \leftarrow \phi + \alpha_\phi \eta_t \Big\{ \big[\frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi} | X_t^{\boldsymbol{\pi}^\phi}) + \zeta_t\big] \Big[\mathrm{d}J(X_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi) \\ + [r(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}) + \gamma p\big(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \boldsymbol{\pi}^\phi(\cdot | X_t^{\boldsymbol{\pi}^\phi})\big) - V]\mathrm{d}t \Big] + \gamma q(X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi}, \phi)\mathrm{d}t \Big\}.$$

Observe the two algorithms above differ by only the presence of the test function η. To illustrate, we describe the second one in Algorithm 3.

5. Applications

In this section we report simulation experiments on our algorithms in two applications. The first one is mean–variance portfolio selection in a finite time horizon with multiple episodes of simulated stock price data. The second application is ergodic linear–quadratic control with a single sample trajectory.

5.1 Mean–Variance Portfolio Selection

We first review the formulation of the exploratory mean–variance portfolio selection problem proposed by Wang and Zhou (2020). The investment universe consists of one risky asset (e.g. a stock index) and one risk-free asset (e.g. a saving account) whose risk-free interest rate is r. The price of the risky asset is governed by a geometric Brownian motion with mean μ and volatility $\sigma > 0$ on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}^W; \{\mathcal{F}_t^W\}_{0 \le t \le T})$:

$$\frac{\mathrm{d}S_t}{S_t} = \mu \mathrm{d}t + \sigma \mathrm{d}W_t. \tag{23}$$

Denote by $\rho = \frac{\mu - r}{\sigma}$ the Sharpe ratio of the risky asset.

An agent has a fixed investment horizon $0 < T < \infty$ and an initial endowment x_0. A self-financing portfolio is represented by the real-valued adapted process $a = \{a_t, 0 \le t \le T\}$, where a_t is the discounted dollar value invested in the risky asset at time t. Then the discounted value of this portfolio satisfies the wealth equation

$$\mathrm{d}x_t^a = a_t[(\mu - r)\mathrm{d}t + \sigma \mathrm{d}W_t] = a_t \frac{\mathrm{d}(e^{-rt}S_t)}{e^{-rt}S_t},\ x_0^a = x_0, \tag{24}$$

where $e^{-rt}S_t$ is the discounted stock price. We stress that the model on the stock price (23) is mainly for theoretical analysis and for generating samples in our simulation; we do not assume that the agent knows its parameters.

The agent has the mean–variance preference, namely, she aims to minimize the variance of the discounted value of the portfolio at T while achieving a given level of expected return:

$$\min_a \mathrm{Var}(x_T^a),\ \text{subject to}\ \ \mathbb{E}[x_T^a] = z, \tag{25}$$

where z is the target value, and the variance and expectation are w.r.t. the probability measure $\mathbb{P}^W$.

Algorithm 3 Actor–Critic Algorithm for Ergodic Tasks

Inputs: initial state x_0, time step Δt, initial learning rates $\alpha_\theta, \alpha_\phi, \alpha_V$ and learning rate schedule function $l(\cdot)$ (a function of time), functional form of the value function $J^\theta(\cdot)$, functional form of the policy $\boldsymbol{\pi}^\phi(\cdot|\cdot)$, functional form of the regularizer $p\big(x, a, \pi(\cdot)\big)$, functional forms of test functions $\boldsymbol{\xi}(x_{\cdot\wedge t}), \boldsymbol{\eta}(x_{\cdot\wedge t}), \boldsymbol{\zeta}(x_{\cdot\wedge t})$, and temperature parameter γ.

Required program: an environment simulator $(x', r) = \mathit{Environment}_{\Delta t}(x, a)$ that takes initial state x and action a as inputs and generates a new state x' (at Δt) and an instantaneous reward r.

Learning procedure:

Initialize θ, ϕ, V. Initialize $k = 0$. Observe the initial state x_0 and store $x_{t_k} \leftarrow x_0$.

loop

Compute test function $\xi_{t_k} = \boldsymbol{\xi}(x_{t_0}, \cdots, x_{t_k})$, $\eta_{t_k} = \boldsymbol{\eta}(x_{t_0}, \cdots, x_{t_k})$ and $\zeta_{t_k} = \boldsymbol{\eta}(x_{t_0}, \cdots, x_{t_k})$.

Generate action $a \sim \boldsymbol{\pi}^\phi(\cdot|x)$.

Apply a to the environment simulator $(x', r) = \mathit{Environment}_{\Delta t}(x, a)$, and observe the output new state x' and reward r. Store $x_{t_{k+1}} \leftarrow x'$.

Compute

$$
\begin{aligned}
&\delta = J^\theta(x') - J^\theta(x) + r\Delta t + \gamma p\big(x, a, \boldsymbol{\pi}^\phi(\cdot|x)\big)\Delta t - V\Delta t,\\
&\Delta\theta = \xi_{t_k}\delta,\\
&\Delta V = \delta,\\
&\Delta\phi = \eta_{t_k}\Big\{\big[\frac{\partial}{\partial\phi}\log \boldsymbol{\pi}^\phi(a|x) + \zeta_{t_k}\big]\delta + \gamma\frac{\partial p}{\partial\phi}\big(x, a, \boldsymbol{\pi}^\phi(\cdot|x)\big)\Delta t\Big\}.
\end{aligned}
$$

Update θ and V (policy evaluation) by

$$\theta \leftarrow \theta + l(k\Delta t)\alpha_\theta\Delta\theta,$$

$$V \leftarrow V + l(k\Delta t)\alpha_V\Delta V.$$

Update ϕ (policy gradient) by

$$\phi \leftarrow \phi + l(k\Delta t)\alpha_\phi\Delta\phi.$$

Update $x \leftarrow x'$ and $k \leftarrow k + 1$.

end loop

This problem is not a standard stochastic control problem and cannot be solved directly by the dynamic programming (DP) principle, or any DP-based reinforcement learning algorithms such as Q-learning. This is because the variance term causes *time-inconsistency* which violates the assumptions of DP. Strotz (1955) discusses three types of agents when facing time-inconsistency. Here, we consider one of them – the so-called *pre-committed* agent who solves the problem at time 0 and sticks to it afterwards.[5] For this type of agent, to overcome the difficulty of DP not directly applicable, Zhou and Li (2000) extend the embedding method, initially introduced by Li and Ng (2000) for the discrete-time mean–variance problem, to transform (25) into an equivalent, unconstrained, and expectation-only problem:

$$\min_a \mathbb{E}[(x_T^a)^2] - z^2 - 2w(\mathbb{E}[x_T^a] - z) = \min_a \mathbb{E}[(x_T^a - w)^2] - (w-z)^2,$$

where w is the Lagrange multiplier associated with the constraint $\mathbb{E}[x_T^a] = z$. This new problem is time-consistent and therefore can be solved by DP. Once the optimal a^* is derived, w can be obtained by the equation $\mathbb{E}[x_T^{a^*}] = z$.

In a reinforcement learning framework, Wang and Zhou (2020) allow randomized actions to incorporate exploration. A stochastic policy is denoted by $\boldsymbol{\pi} = \boldsymbol{\pi}(\cdot|t,x)$, namely, at any current time–wealth pair (t,x), the total amount of discounted wealth invested in the stock is a random draw from the distribution with the density function $\boldsymbol{\pi}(\cdot|t,x)$. Under such a policy, we denote by $\tilde{X}^{\boldsymbol{\pi}} = \{\tilde{X}_s^{\boldsymbol{\pi}} : t \le s \le T\}$ the solution to the following SDE

$$\mathrm{d}\tilde{X}_s^{\boldsymbol{\pi}} = (\mu - r)\int_{\mathbb{R}} a\boldsymbol{\pi}(a|s,\tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}s + \sigma\sqrt{\int_{\mathbb{R}} a^2\boldsymbol{\pi}(a|s,\tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a}\mathrm{d}W_s;\ \tilde{X}_t^{\boldsymbol{\pi}} = x,$$

which is (5) specializing to the current case.

Moreover, an entropy regularizer is added to incentivize exploration. Mathematically, the entropy-regularized mean–variance portfolio choice problem is to solve

$$V(t,x;w) = \min_{\boldsymbol{\pi}} \mathbb{E}\left[(\tilde{X}_T^{\boldsymbol{\pi}} - w)^2 - \gamma\int_t^T \mathcal{H}(\pi_s)\mathrm{d}s\middle|\tilde{X}_t^{\boldsymbol{\pi}} = x\right] - (w-z)^2, \tag{26}$$

where z is the target expected terminal wealth, $\pi_s = \boldsymbol{\pi}(\cdot|s,\tilde{X}_s^{\boldsymbol{\pi}}),\ t \le s \le T$, $\mathcal{H}$ is the differential entropy $\mathcal{H}(\pi) = -\int_{\mathcal{A}} \pi(a)\log\pi(a)\mathrm{d}a$, γ is the temperature parameter, and w is the Lagrange multiplier similar to that introduced earlier.

We follow Wang and Zhou (2020) to parameterize the value function by

$$J^\theta(t,x;w) = (x-w)^2 e^{-\theta_3(T-t)} + \theta_2(t^2 - T^2) + \theta_1(t-T) - (w-z)^2,$$

and parameterize the policy by

$$\boldsymbol{\pi}^\phi(\cdot|t,x;w) = \mathcal{N}(\cdot|-\phi_1(x-w), e^{\phi_2+\phi_3(T-t)}),$$

where $\mathcal{N}(\cdot|\alpha,\delta^2)$ is the pdf of the normal distribution with mean α and variance δ^2. These function approximators are derived in Wang and Zhou (2020) by exploiting the special structure of the underlying problem.

5. The other two types are the *naïve* one who re-optimizes at any given time and the *sophisticated* one who seeks subgame perfect Nash equilibria among her-selves at different times. The latter has been well studied in the continuous-time setting in recent years; see e.g. Ekeland and Lazrak (2006); Björk et al. (2014); Basak and Chabakauri (2010); Dai et al. (2021). The RL counterpart is studied in Dai et al. (2020).

There is no running reward from the actions except the regularizer

$$\mathcal{H}(\boldsymbol{\pi}^{\phi}(\cdot|t,x;w)) = -\frac{1}{2}\log(2\pi e) - \frac{1}{2}[\phi_2 + \phi_3(T-t)] =: \hat{p}(t,\phi).$$

Note that the regularizer turns out to be independent of the state x. Finally, the discount factor is $\beta = 0$.

From this point on, we depart from Wang and Zhou (2020) and instead apply the methods developed in this paper to solve the problem. We choose the test functions for PE as the following gradients, in accordance with the most popular $TD(0)$ algorithm:[6]

$$\frac{\partial J^{\theta}}{\partial \theta_1}(t,x;w) = t - T,\;\; \frac{\partial J^{\theta}}{\partial \theta_2}(t,x;w) = t^2 - T^2,\;\; \frac{\partial J^{\theta}}{\partial \theta_3}(t,x;w) = (x-w)^2 e^{-\theta_3(T-t)}(t-T).$$

The PE updating rule is

$$\theta \leftarrow \theta + \alpha_\theta \int_0^T \frac{\partial J^{\theta}}{\partial \theta}(t, X_t^{\boldsymbol{\pi}^{\phi}}; w)\left[\mathrm{d}J^{\theta}(t, X_t^{\boldsymbol{\pi}^{\phi}}; w) + \gamma \hat{p}(t,\phi)\mathrm{d}t\right].$$

For the PG part, the gradients of log-likelihood are

$$\frac{\partial \log \boldsymbol{\pi}^{\phi}(a|t,x;w)}{\partial \phi_1} = -\big(a + \phi_1(x-w)\big)(x-w)e^{-\phi_2-\phi_3(T-t)},$$

$$\frac{\partial \log \boldsymbol{\pi}^{\phi}(a|t,x;w)}{\partial \phi_2} = -\frac{1}{2} + \frac{\big(a+\phi_1(x-w)\big)^2}{2}e^{-\phi_2-\phi_3(T-t)},$$

$$\frac{\partial \log \boldsymbol{\pi}^{\phi}(a|t,x;w)}{\partial \phi_3} = -\frac{T-t}{2} + \frac{\big(a+\phi_1(x-w)\big)^2}{2}e^{-\phi_2-\phi_3(T-t)}(T-t),$$

and those of the regularizer are

$$\frac{\partial \hat{p}}{\partial \phi_1}(t,\phi) = 0,\;\; \frac{\partial \hat{p}}{\partial \phi_2}(t,\phi) = -\frac{1}{2},\;\; \frac{\partial \hat{p}}{\partial \phi_3}(t,\phi) = -\frac{T-t}{2}.$$

Accordingly, the PG updating rule is

$$\phi \leftarrow \phi - \alpha_\phi \int_0^T \left\{\frac{\partial \log \boldsymbol{\pi}^{\phi}}{\partial \phi}(a_t|t, X_t^{\boldsymbol{\pi}^{\phi}}; w)\left[\mathrm{d}J^{\theta}(t, X_t^{\boldsymbol{\pi}^{\phi}}; w) + \gamma\hat{p}(t, X_t^{\boldsymbol{\pi}^{\phi}}, \phi)\mathrm{d}t\right] + \gamma\frac{\partial \hat{p}}{\partial \phi}(t, X_t^{\boldsymbol{\pi}^{\phi}}, \phi)\mathrm{d}t\right\}.$$

In addition, there is the Lagrange multiplier w we need to learn: we update w based on the same stochastic approximation scheme in Wang and Zhou (2020).

We present our algorithm as Algorithm 4. Then we replicate the simulation study of Wang and Zhou (2020) with the same basic setting: $x_0 = 1$, $z = 1.4$, $T = 1$, $\Delta t = \frac{1}{252}$. Choose temperature parameter $\gamma = 0.1$, training sample size $N = 2 \times 10^6$, and batch size $m = 10$ for updating the Lagrange multiplier. The learning rate parameters in Wang and

6. Wang and Zhou (2020) employ a mean–square TD error (MSTDE) algorithm to do PE and a policy improvement theorem to update policies. However, it is shown in Jia and Zhou (2021) that MSTDE only minimizes the qudratic variation of the martingale, which may not lead to the true solution of PE. As discussed earlier, other PE algorithms proposed in Jia and Zhou (2021) can also be applied.

Zhou (2020) are set to be $\alpha_w = 0.05$, and $\alpha_\theta = \alpha_\phi = 0.0005$ without decay (i.e. $l(j) \equiv 1$). In our experiment we adopt these learning rate values for the Wang and Zhou (2020) algorithm unless the algorithm does not converge, in which case we tune the initial learning rates or introduce and tune the decay rate to guarantee convergence. For our algorithm, we set $\alpha_w = 0.05$, and $\alpha_\theta = \alpha_\phi = 0.1$ with decay rate $l(j) = j^{-0.51}$ and tune the initial learning rate when necessary. The initialization of the parameters θ and ϕ is set to be all 0 for both algorithms (the initialization is not discussed in Wang and Zhou 2020). The training set is fixed, with N independent 1-year simulated stock price trajectories. Instead of examining the in-sample performance, we test the learned policies of both methods on an *independent* test set consisting of $B = 10^5$ independent 1-year simulated stock price trajectories drawn from the same distribution as the training set.[7] We then compute the mean and variance of the values of the discounted terminal wealth on the test set, along with the Sharpe ratio defined as $\frac{\text{mean}-1}{\sqrt{\text{variance}}}$.

Table 1 presents the test results when stock price is generated from geometric Brownian motion under different specifications of the market parameters μ and σ. Our algorithm achieves higher Sharpe ratios in all the scenarios. In particular, when the theoretically optimal Sharpe ratios are high, our algorithm performs much better than Wang and Zhou (2020) out-of-sample, even if its Sharpe ratios are still much lower than the optimal ones. On the other hand, when the theoretical optimal Sharpe ratios are low, our algorithm's achieved Sharpe ratios are close to optimal and the mean returns are close to the target ones as well.

5.2 Ergodic Linear–Quadratic Control

Consider the ergodic linear–quadratic (LQ) control problem where state responds to actions in a linear way

$$\mathrm{d}X_t = (AX_t + Ba_t)\mathrm{d}t + (CX_t + Da_t)\mathrm{d}W_t,\ X_0 = x_0,$$

and the goal is to maximize the long term average payoff

$$\liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\int_0^T r(X_t, a_t)\mathrm{d}t | X_0 = x_0\right],$$

with $r(x,a) = -(\frac{M}{2}x^2 + Rxa + \frac{N}{2}a^2 + Px + Qa)$.

In the entropy-regularized RL formulation, the policy is denoted by $\boldsymbol{\pi}(\cdot|x)$ and actions are generated from this policy. The corresponding goal is to maximize

$$\liminf_{T\to\infty} \frac{1}{T}\mathbb{E}^{\mathbb{P}^W}\left[\int_0^T \int_{\mathbb{R}} r(\tilde{X}_t^{\boldsymbol{\pi}}, a)\boldsymbol{\pi}(a|\tilde{X}_t^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}t + \gamma\mathcal{H}(\boldsymbol{\pi}(\,\cdot\,|\tilde{X}_t^{\boldsymbol{\pi}}))\mathrm{d}t\Big|\tilde{X}_0^{\boldsymbol{\pi}} = x_0\right],$$

where $\mathcal{H}$ is the differential entropy as before. Moreover, $\tilde{X}^{\boldsymbol{\pi}}$ satisfies

$$\mathrm{d}\tilde{X}_t^{\boldsymbol{\pi}} = \int_{\mathbb{R}} (A\tilde{X}_t^{\boldsymbol{\pi}} + Ba)\boldsymbol{\pi}(a|\tilde{X}_t^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}t + \sqrt{\int_{\mathbb{R}} \left(C\tilde{X}_t^{\boldsymbol{\pi}\,2} + Da\right)^2\boldsymbol{\pi}(a|\tilde{X}_t^{\boldsymbol{\pi}})\mathrm{d}a}\mathrm{d}W_t.$$

7. Wang and Zhou (2020) report in-sample performance of the last 2000 iterations in the training set but does not present out-of-sample test results.

Algorithm 4 Offline–Episodic Actor–Critic Mean–Variance Algorithm

Inputs: initial state x_0, horizon T, time step Δt, number of episodes N, number of time grids K, initial learning rates $\alpha_\theta, \alpha_\phi, \alpha_w$ and learning rate schedule function $l(\cdot)$ (a function of the number of episodes), and temperature parameter γ.

Required program: a market simulator $x' = Market_{\Delta t}(t, x, a)$ that takes current time-state pair (t, x) and action a as inputs and generates state x' at time $t + \Delta t$.

Learning procedure:

Initialize θ, ϕ, w.
for episode $j = 1$ **to** N **do**
Initialize $k = 0$. Observe the initial state x and store $x_{t_k} \leftarrow x$.
while $k < K$ **do**
Compute and store the test function $\xi_{t_k} = \frac{\partial J^\theta}{\partial \theta}(t_k, x_{t_k}; w)$.
Generate action $a_{t_k} \sim \boldsymbol{\pi}^\phi(\cdot|t_k, x_{t_k})$.
Apply a_{t_k} to the market simulator $x = Market_{\Delta t}(t_k, x_{t_k}, a_{t_k})$, and observe the output new state x. Store $x_{t_{k+1}}$.
Update $k \leftarrow k + 1$.
end while
Store the terminal wealth $X_T^{(j)} \leftarrow x_{t_K}$.
Compute

$$\Delta\theta = \sum_{i=0}^{K-1} \xi_{t_i} \big[J^\theta(t_{i+1}, x_{t_{i+1}}; w) - J^\theta(t_i, x_{t_i}; w) + \gamma \hat{p}(t_i, x_{t_i}, \phi)\Delta t \big],$$

$$\Delta\phi = \sum_{i=0}^{K-1} \frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^\phi(a_{t_i}|t_i, x_{t_i}) \big[J^\theta(t_{i+1}, x_{t_{i+1}}) - J^\theta(t_i, x_{t_i}) + \gamma \hat{p}(t, x_{t_i}, \phi)\Delta t \big] + \gamma \frac{\partial \hat{p}}{\partial \phi}(t_i, x_{t_i}, \phi)\Delta t.$$

Update θ (policy evaluation) by

$$\theta \leftarrow \theta + l(j)\alpha_\theta \Delta\theta.$$

Update ϕ (policy gradient) by

$$\phi \leftarrow \phi - l(j)\alpha_\phi \Delta\phi.$$

Update w (Lagrange multiplier) every m episodes:
if $j \equiv 0 \mod m$ **then**

$$w \leftarrow w - \alpha_w \frac{1}{m} \sum_{i=j-m+1}^{j} X_T^{(i)}.$$

end if
end for

Table 1: **Out-of-sample performance comparison in terms of mean, variance and Sharpe ratio when data is generated by geometric Brownian motion.** The first two columns are market configurations and the third column is the theoretically optimal Sharpe ratio. The learned policies are tested on an independent test set with 10^5 trajectories and we report the mean, variance of the terminal wealth and then compute the Sharpe ratio. We compare our Sharpe ratio with that produced by the algorithm in Wang and Zhou (2020) and highlight the larger one in **bold**.

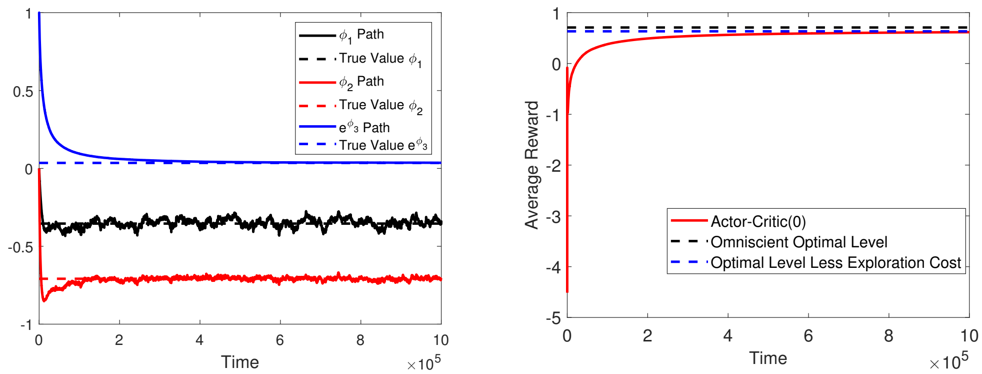
Parameters		Optimal	This paper			Wang and Zhou (2020)		
μ	σ	Sharpe ratio	mean	variance	Sharpe ratio	mean	variance	Sharpe ratio
-0.5	0.1	744151.56	1.401	0.000	**72.292**	1.400	0.004	6.454
-0.3	0.1	167.332	1.401	0.001	**14.041**	1.399	0.011	3.829
-0.1	0.1	1.795	1.396	0.052	**1.735**	1.399	0.090	1.334
0	0.1	0.202	1.373	3.406	**0.202**	1.440	4.944	0.198
0.1	0.1	0.947	1.423	0.200	**0.946**	1.409	0.221	0.870
0.3	0.1	50.391	1.404	0.002	**9.782**	1.404	0.015	3.320
0.5	0.1	100709.96	1.401	0.000	**52.620**	1.403	0.005	5.920
-0.5	0.2	29.354	1.402	0.001	**15.853**	1.399	0.014	3.411
-0.3	0.2	3.455	1.399	0.014	**3.350**	1.398	0.041	1.978
-0.1	0.2	0.658	1.390	0.351	**0.658**	1.402	0.378	0.653
0	0.2	0.100	1.363	13.088	**0.100**	1.144	2.060	0.100
0.1	0.2	0.417	1.427	1.052	**0.417**	1.396	0.967	0.403
0.3	0.2	2.470	1.406	0.028	**2.422**	1.408	0.057	1.710
0.5	0.2	17.786	1.400	0.001	**11.529**	1.405	0.017	3.126
-0.5	0.3	4.379	1.401	0.009	**4.345**	1.398	0.029	2.338
-0.3	0.3	1.456	1.395	0.074	**1.451**	1.398	0.091	1.317
-0.1	0.3	0.417	1.369	0.784	**0.416**	1.385	0.900	0.406
0	0.3	0.067	1.362	29.379	**0.067**	1.525	61.832	0.067
0.1	0.3	0.271	1.425	2.455	**0.271**	1.118	0.197	0.267
0.3	0.3	1.179	1.418	0.126	**1.178**	1.417	0.135	1.136
0.5	0.3	3.455	1.404	0.015	**3.278**	1.408	0.036	2.139
-0.5	0.4	2.102	1.398	0.038	**2.035**	1.399	0.051	1.764
-0.3	0.4	0.947	1.392	0.172	**0.946**	1.404	0.187	0.935
-0.1	0.4	0.307	1.378	1.519	**0.307**	1.382	1.585	0.304
0	0.4	0.050	1.368	54.013	**0.050**	1.589	138.806	0.050
0.1	0.4	0.202	1.429	4.507	**0.202**	1.162	0.653	0.200
0.3	0.4	0.795	1.426	0.287	**0.795**	1.391	0.273	0.748
0.5	0.4	1.795	1.412	0.054	**1.767**	1.413	0.066	1.612

Following the same line of deductions as in Wang et al. (2020), we can show that the optimal policy is a normal distribution whose mean is linear in state and variance a constant. Therefore we parameterize the policy by $\boldsymbol{\pi}^{\phi}(\cdot|x) = \mathcal{N}(\cdot|\phi_1 x + \phi_2, e^{\phi_3})$. Moreover, the function J is parameterized as a quadratic function $J^{\theta}(x) = \frac{1}{2}\theta_0 x^2 + \theta_1 x$ (we ignore the constant term since J is unique up to a constant) and the optimal value V is an extra parameter.

This problem falls into the formulation of an ergodic task; so we directly implement Algorithm 3 in our simulation and then compare the learned parameters with the theoretically optimal ones. In addition, we compare the up-to-now average reward during the online learning process to two theoretical benchmarks. The first one is the omniscient optimal level, which is the maximum long term average reward that can be achieved by a hypothetical agent who knows completely about the environment (i.e. the correct model and model parameters) and acts optimally (the optimal policy is a deterministic one) without needing to explore (and hence there is no entropy regularization). The second benchmark is the omniscient optimal level less the exploration cost, which is the maximum long term average reward that can be achieved by the aforementioned hypothetical agent who is however forced to explore under entropy regularization. Clearly, since exploration (rendering a stochastic policy) is inherent in the RL setting, our algorithm can at most achieve the second benchmark. In other words, after learning for a sufficiently long time, we can learn the correct optimal policy but can only expect the up-to-now average reward to approach the optimal level less the exploration cost.

To guarantee the stationarity of the controlled state process, we set $A = -1, B = C = 0$ and $D = 1$. Moreover, we set $x_0 = 0$, $M = N = Q = 2$, $R = P = 1$, and $\gamma = 0.1$. Learning rate is initialized as $\alpha_{\theta} = \alpha_{\phi} = 0.001$, and decays according to $l(t) = \frac{1}{\max\{1, \log t\}}$. All the parameters to be learned are initialized as 0 and time discretization is taken as $\Delta t = 0.01$.

We implement $TD(0)$ for both the PE and the PG parts of the AC algorithm, referred to as Actor–Critic(0) algorithm in Figure 1. Namely, we choose test functions $\xi_t = \frac{\partial J^{\theta}}{\partial \theta}(X_t), \eta_t = 1, \zeta_t = 0$ in Algorithm 3. Figure 1 shows the convergence of the learned policy parameters along with that of the average reward along a single state sample trajectory. Observe that the average reward first decreases at the beginning of this particular trajectory. The reason may have been that during the initial iterations the underlying state process has not yet converged to the stationary distribution and the initial policies are still far away from the optimal one, and hence the average reward is dominated by a few “wrong trials”. After a sufficient amount of time, however, both the policies and the average reward start to converge to the theoretically optimal values. Between the two it takes a much longer time for the average reward to approach the optimal level as we wait for the contribution from the bad performance of the beginning period to diminish.



(a) The learned parameters in the policy along one sample trajectory.

(b) The average reward along one sample trajectory.

Figure 1: **Convergence of the learned policy and the average reward under the online learning algorithm.** A single state trajectory is generated with length $T = 10^6$ under the online AC algorithm. The left panel illustrates the convergence of the policy parameters, where the dashed horizontal lines indicate the values of the respective parameters of the theoretically optimal policy to the entropy-regularized exploratory stochastic control problem. The right panel shows the convergence of the average reward, where the two dashed horizontal lines are respectively the omniscient optimal average reward without exploration when the model parameters are known, and the omniscient optimal average reward less the exploration cost.

6. Conclusion

This paper is the final installment of a “trilogy”, the first two being Wang et al. (2020) and Jia and Zhou (2021), that endeavors to develop a systematic and unified theoretical foundation for RL in continuous time with continuous state space and possibly continuous action space. The previous two papers address exploration and PE respectively, and this paper focuses on PG. A major finding of the current paper is that PG is intimately related to PE, and thus the martingale characterization of PE established in Jia and Zhou (2021) can be applied to PG. Combining the theoretical results of the three papers, we propose online and offline actor–critic algorithms for general model-free RL tasks, where we learn value functions and stochastic policies simultaneously and alternatingly.

This series of papers are characterized by conducting all the theoretical analysis within the continuous setting and discretizing time only when implementing the algorithms. The advantages of this approach, as versus discretizing time right at the start and then applying existing MDP results, are articulated in Doya (2000). Moreover, there are more analytical tools at our disposal in the continuous setting, including calculus, stochastic calculus, stochastic control, and differential equations. The discrete-time versions of the various algorithms devised in the three papers are indeed well known in the discrete-time RL literature; hence their convergence is well established. On the other hand, Jia and Zhou (2021) prove that any convergent time-discretized PE algorithm also converges as the mesh size goes to zero. Because PG algorithms developed in the current paper are essentially derived from the martingality for PE, the same convergence also holds for them.

It is interesting to note that the derivation and representation of PG is not entirely analogous to that for MDPs. The latter involves state–action function (Q-function), whereas the former is essentially the expected integration of a term involving the value function. Also, PG results in MDPs often assume the stationary distribution exists, while our PG representation for the continuous setting does not need such an assumption.

The study on continuous-time RL is still in its infancy, and open questions abound. These include, to name but a few, regret bound of episodic RL problems in terms of the length of the episode, interpretation of Q-function and Q-learning in the continuous setting, and dependence of the performance of an AC algorithm on the temperature parameter when there is an exploration regularizer.

Acknowledgments

Zhou gratefully acknowledges financial support through a start-up grant and the Nie Center for Intelligent Asset Management at Columbia University.

Appendix. Proofs of Statements

In all the proofs we use generic notations $C_1, C_2, \cdots$ to denote constants that are independent of other variables involved such as t, x, a. A same such notation may show up in different places but does not necessarily have the same value.

Proof of Lemma 2

We start by examining $\tilde{b}\big(t, x, \boldsymbol{\pi}(\cdot|t,x)\big)$. Note that

$$
\begin{aligned}
&\tilde{b}\big(t, x, \boldsymbol{\pi}(\cdot|t,x)\big) - \tilde{b}\big(t, x', \boldsymbol{\pi}(\cdot|t,x')\big) \\
=&\int_{\mathcal{A}} [b(t,x,a) - b(t,0,a)][\boldsymbol{\pi}(a|t,x) - \boldsymbol{\pi}(a|t,x')]\mathrm{d}a + \int_{\mathcal{A}} b(t,0,a)[\boldsymbol{\pi}(a|t,x) - \boldsymbol{\pi}(a|t,x')]\mathrm{d}a \\
&+ \int_{\mathcal{A}} [b(t,x,a) - b(t,x',a)]\boldsymbol{\pi}(a|t,x')\mathrm{d}a.
\end{aligned}
$$

Hence

$$
\begin{aligned}
&\Big|\tilde{b}\big(t, x, \boldsymbol{\pi}(\cdot|t,x)\big) - \tilde{b}\big(t, x', \boldsymbol{\pi}(\cdot|t,x')\big)\Big| \\
\leq&\int_{\mathcal{A}} |b(t,x,a) - b(t,0,a)||\boldsymbol{\pi}(a|t,x) - \boldsymbol{\pi}(a|t,x')|\mathrm{d}a + \int_{\mathcal{A}} |b(t,0,a)||\boldsymbol{\pi}(a|t,x) - \boldsymbol{\pi}(a|t,x')|\mathrm{d}a \\
&+ \int_{\mathcal{A}} |b(t,x,a) - b(t,x',a)|\boldsymbol{\pi}(a|t,x')\mathrm{d}a \\
\leq&(C_1|x| + C_2)\int_{\mathcal{A}} |\boldsymbol{\pi}(a|t,x) - \boldsymbol{\pi}(a|t,x')|\mathrm{d}a + C_1|x - x'| \\
\leq&C_1|x - x'| + (C_1|x| + C_2)C_3|x - x'|.
\end{aligned}
$$

Moreover, note that $|\tilde{b}\big(t, x, \boldsymbol{\pi}(\cdot|t,x)\big)| \leq \int_{\mathcal{A}} |b(t,x,a)|\boldsymbol{\pi}(a|t,x) \leq \int_{\mathcal{A}} (C_1|x| + C_2)\boldsymbol{\pi}(a|t,x) = C_1|x| + C_2$.

Similarly, we can show that $\tilde{\sigma}\big(t, x, \boldsymbol{\pi}(\cdot|t,x)\big)$ is locally Lipschitz continuous and has linear growth in x. The unique existence of the strong solution to (5) then follows from the standard SDE theory.

Next, the SDE (5) yields

$$
\tilde{X}_s^{\boldsymbol{\pi}} = x + \int_t^s \tilde{b}\big(\tau, \tilde{X}_\tau^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\cdot|\tau, \tilde{X}_\tau^{\boldsymbol{\pi}})\big)\mathrm{d}\tau + \int_t^s \tilde{\sigma}\big(\tau, \tilde{X}_\tau^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\cdot|\tau, \tilde{X}_\tau^{\boldsymbol{\pi}})\big)\mathrm{d}W_\tau.
$$

Based on the proved growth condition on $\tilde{b}, \tilde{\sigma}$, CauchySchwarz inequality, and Burkholder-Davis-Gundy inequalities, we obtain

$$
\begin{aligned}
&\mathbb{E}\Big[\max_{t\le s\le T'} |\tilde{X}_s^{\boldsymbol{\pi}}|^\mu \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x\Big] \\
\le & C_1 \mathbb{E}\Big[|x|^\mu + \max_{t\le s\le T'} \Big| \int_t^s \tilde{b}\big(\tau, \tilde{X}_\tau^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\cdot|\tau, \tilde{X}_\tau^{\boldsymbol{\pi}})\big)\mathrm{d}\tau \Big|^\mu \\
&+ \max_{t\le s\le T'} \Big| \int_t^s \tilde{\sigma}\big(\tau, \tilde{X}_\tau^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\cdot|\tau, \tilde{X}_\tau^{\boldsymbol{\pi}})\big)\mathrm{d}W_\tau \Big|^\mu \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x\Big] \\
\le & C_1 \mathbb{E}\Big[|x|^\mu + C_2 \int_t^{T'} (1 + \max_{t\le s\le \tau} |\tilde{X}_s^{\boldsymbol{\pi}}|)^\mu \mathrm{d}\tau + C_3 \big(\int_t^{T'} (1 + \max_{t\le s\le \tau} |\tilde{X}_s^{\boldsymbol{\pi}}|)^2 \mathrm{d}\tau\big)^{\mu/2} \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x\Big] \\
\le & C_4 (1 + |x|^\mu) + C_5 \int_t^{T'} \mathbb{E}\Big[\max_{t\le s\le \tau} |\tilde{X}_s^{\boldsymbol{\pi}}|^\mu \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x\Big]\mathrm{d}\tau.
\end{aligned}
$$

Applying Gronwall's inequality to $\mathbb{E}\Big[\max_{t\le s\le T'} |\tilde{X}_s^{\boldsymbol{\pi}}|^\mu \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x\Big]$ as a function of T', we obtain the second desired result of the lemma. The final result is evident.

Proof of Lemma 3

The proof follows from applying Beck et al. (2021, Theorem 1.1) to the SDE (5): under Assumption 1 along with Definition 1, Lemma 2 verifies the sufficient conditions in Beck et al. (2021).

Proof of Theorem 4

The first statement follows from Jia and Zhou (2021, Proposition 1) along with the Markov property of the solution to the SDE (5).

For the second statement, according to Jia and Zhou (2021, Proposition 4), we have the following martingale orthogonality condition for $\tilde{X}^{\boldsymbol{\pi}}$:

$$
\begin{aligned}
\mathbb{E}^{\mathbb{P}^W}\Big[\int_0^T \xi_t \big[\mathrm{d}J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - \beta J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi})\mathrm{d}t \\
+ \int_{\mathcal{A}} [r(t, \tilde{X}_t^{\boldsymbol{\pi}}, a) + \gamma p\big(t, \tilde{X}_t^{\boldsymbol{\pi}}, a, \boldsymbol{\pi}(\cdot|t, \tilde{X}_t^{\boldsymbol{\pi}})\big)]\boldsymbol{\pi}(a|t, \tilde{X}_t^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}t\big] \Big| \tilde{X}_0^{\boldsymbol{\pi}} = x\Big] = 0
\end{aligned}
$$

for all $\xi \in L^2_{\mathcal{F}^{\tilde{X}^{\boldsymbol{\pi}}}}\big([0,T]; J(\cdot, \tilde{X}_\cdot^{\boldsymbol{\pi}}; \boldsymbol{\pi})\big)$. Now, any $\xi \in L^2_{\mathcal{F}^{\tilde{X}^{\boldsymbol{\pi}}}}\big([0,T]; J(\cdot, \tilde{X}_\cdot^{\boldsymbol{\pi}}; \boldsymbol{\pi})\big)$ corresponds to a measurable functional $\boldsymbol{\xi} : [0,T] \times C([0,T]; \mathbb{R}^d) \mapsto \mathbb{R}$ such that $\xi_t = \boldsymbol{\xi}(t, \tilde{X}_{t\wedge\cdot}^{\boldsymbol{\pi}})$. However, $\tilde{X}^{\boldsymbol{\pi}}$ and $X^{\boldsymbol{\pi}}$ have the same distribution and $a_t^{\boldsymbol{\pi}} \sim \boldsymbol{\pi}(\cdot|t, \tilde{X}_t^{\boldsymbol{\pi}})$; hence

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}^W}\Big[\int_0^T \boldsymbol{\xi}(t, \tilde{X}_{t\wedge\cdot}^{\boldsymbol{\pi}}) \big[\mathrm{d}J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - \beta J(t, \tilde{X}_t^{\boldsymbol{\pi}}; \boldsymbol{\pi})\mathrm{d}t\big] \Big| \tilde{X}_0^{\boldsymbol{\pi}} = x\Big] \\
=&\mathbb{E}^{\mathbb{P}}\Big[\int_0^T \boldsymbol{\xi}(t, X_{t\wedge\cdot}^{\boldsymbol{\pi}}) \big[\mathrm{d}J(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - \beta J(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi})\mathrm{d}t\big] \Big| X_0^{\boldsymbol{\pi}} = x\Big],
\end{aligned}
$$

and

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}^W}\bigg[\int_0^T \boldsymbol{\xi}(t,\tilde{X}^{\boldsymbol{\pi}}_{t\wedge\cdot})\int_{\mathcal{A}}[r(t,\tilde{X}^{\boldsymbol{\pi}}_t,a)+\gamma p\big(t,\tilde{X}^{\boldsymbol{\pi}}_t,a,\boldsymbol{\pi}(\cdot|t,\tilde{X}^{\boldsymbol{\pi}}_t)\big)]\boldsymbol{\pi}(a|t,\tilde{X}^{\boldsymbol{\pi}}_t)\mathrm{d}a\mathrm{d}t\Big|\tilde{X}^{\boldsymbol{\pi}}_0=x\bigg]\\
=&\mathbb{E}^{\mathbb{P}}\bigg[\int_0^T \boldsymbol{\xi}(t,X^{\boldsymbol{\pi}}_{t\wedge\cdot})[r(t,X^{\boldsymbol{\pi}}_t,a^{\boldsymbol{\pi}}_t)+\gamma p\big(t,X^{\boldsymbol{\pi}}_t,a^{\boldsymbol{\pi}}_t,\boldsymbol{\pi}(\cdot|t,X^{\boldsymbol{\pi}}_t)\big)]\mathrm{d}t\Big|X^{\boldsymbol{\pi}}_0=x\bigg].
\end{aligned}
$$

Combining the above two equations leads to (9).

Proof of Theorem 5

It suffices to prove (12) equals (14).

Fix t. Define a sequence of stopping times $\tau_n=\inf\{s\geq t:|X^{\boldsymbol{\pi}^\phi}_s|\geq n\}$. Applying Itô's lemma to $J(s,X^{\boldsymbol{\pi}^\phi}_s)$, we obtain:

$$
\begin{aligned}
&\int_t^{T\wedge\tau_n} e^{-\beta(s-t)}\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a^{\boldsymbol{\pi}^\phi}_s|s,X^{\boldsymbol{\pi}^\phi}_s)\big]\times\big[\mathrm{d}J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)\\
&+[r(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s)+\gamma p\big(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\boldsymbol{\pi}^\phi(\cdot|s,X^{\boldsymbol{\pi}^\phi}_s)\big)-\beta J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)]\mathrm{d}s\big]+\gamma q(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\phi)\mathrm{d}s\Big\}\\
=&\int_t^{T\wedge\tau_n} e^{-\beta(s-t)}\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a^{\boldsymbol{\pi}^\phi}_s|s,X^{\boldsymbol{\pi}^\phi}_s)\big]\times\big\{\big[\mathcal{L}^{a_s}J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)+r(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s)\\
&+\gamma p\big(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\boldsymbol{\pi}^\phi(\cdot|s,X^{\boldsymbol{\pi}^\phi}_s)\big)-\beta J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)\big]\mathrm{d}s+\frac{\partial J}{\partial x}(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)^\top\sigma(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s)\mathrm{d}W_s\big\}\\
&+\gamma q(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\phi)\mathrm{d}s\Big\}.
\end{aligned}
$$

Taking expectation yields

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}}\bigg[\int_t^{T\wedge\tau_n} e^{-\beta(s-t)}\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a^{\boldsymbol{\pi}^\phi}_s|s,X^{\boldsymbol{\pi}^\phi}_s)\big]\times\big[\mathrm{d}J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)\\
&+[r(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s)+\gamma p\big(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\boldsymbol{\pi}^\phi(\cdot|s,X^{\boldsymbol{\pi}^\phi}_s)\big)-\beta J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)]\mathrm{d}s\big]+\gamma q(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\phi)\mathrm{d}s\Big\}\Big|X^{\boldsymbol{\pi}^\phi}_t=x\bigg]\\
=&\mathbb{E}^{\mathbb{P}}\bigg[\int_t^{T\wedge\tau_n} e^{-\beta(s-t)}\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a^{\boldsymbol{\pi}^\phi}_s|s,X^{\boldsymbol{\pi}^\phi}_s)\big]\times\big\{\big[\mathcal{L}^{a^{\boldsymbol{\pi}^\phi}_s}J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)+r(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s)\\
&+\gamma p\big(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\boldsymbol{\pi}^\phi(\cdot|s,X^{\boldsymbol{\pi}^\phi}_s)\big)-\beta J(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)\big]\mathrm{d}s\big\}+\gamma q(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s,\phi)\mathrm{d}s\Big\}\Big|X^{\boldsymbol{\pi}^\phi}_t=x\bigg]\\
&+\mathbb{E}^{\mathbb{P}}\bigg[\int_t^{T\wedge\tau_n} e^{-\beta(s-t)}\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^\phi(a^{\boldsymbol{\pi}^\phi}_s|s,X^{\boldsymbol{\pi}^\phi}_s)\big]\frac{\partial J}{\partial x}(s,X^{\boldsymbol{\pi}^\phi}_s;\boldsymbol{\pi}^\phi)^\top\sigma(s,X^{\boldsymbol{\pi}^\phi}_s,a_s)\mathrm{d}W_s\Big|X^{\boldsymbol{\pi}^\phi}_t=x\bigg]\\
=&\mathbb{E}^{\mathbb{P}}\bigg[\int_t^{T\wedge\tau_n} e^{-\beta(s-t)}\check{r}(s,X^{\boldsymbol{\pi}^\phi}_s,a^{\boldsymbol{\pi}^\phi}_s;\phi)\mathrm{d}s\Big|X^{\boldsymbol{\pi}^\phi}_t=x\bigg].
\end{aligned}
$$

The second term above vanishes because when $t \le s \le T \wedge \tau_n$, it follows from Assumptions 1 that

$$
\begin{aligned}
&\Big| \int_{\mathcal{A}} \big[\frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}})\big] \times \frac{\partial J}{\partial x}(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi})^{\top} \sigma(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}, a) \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a \Big|^2 \\
\le& \int_{\mathcal{A}} \Big| \frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \frac{\partial J}{\partial x}(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi})^{\top} \sigma(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}, a) \Big|^2 \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a \\
\le& C \max_{|x| \le n, 0 \le t \le T} \big| \frac{\partial}{\partial x} J(t, x; \boldsymbol{\pi}^{\phi}) \big| \int_{\mathcal{A}} \Big| \frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \Big|^2 (1 + |\tilde{X}_s^{\boldsymbol{\pi}^{\phi}}|)^2 \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a \\
\le& C(1+n)^2 \max_{|x| \le n, 0 \le t \le T} \big| \frac{\partial}{\partial x} J(t, x; \boldsymbol{\pi}^{\phi}) \big| \int_{\mathcal{A}} \Big| \frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \Big|^2 \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a,
\end{aligned}
$$

which is bounded by a function of n due to Assumption 3. Thus,

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}} \Big[\int_t^{T \wedge \tau_n} e^{-\beta(s-t)} \big[\frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^{\phi}(a_s^{\boldsymbol{\pi}^{\phi}}|s, X_s^{\boldsymbol{\pi}^{\phi}})\big] \frac{\partial J}{\partial x}(s, X_s^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi})^{\top} \sigma(s, X_s^{\boldsymbol{\pi}^{\phi}}, a_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}W_s \Big| X_t^{\boldsymbol{\pi}^{\phi}} = x \Big] \\
=& \mathbb{E}^{\mathbb{P}^W} \Big[\int_t^{T \wedge \tau_n} e^{-\beta(s-t)} \int_{\mathcal{A}} \big[\frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}})\big] \\
&\times \frac{\partial J}{\partial x}(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}; \boldsymbol{\pi}^{\phi})^{\top} \sigma(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}, a) \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a \mathrm{d}W_s \Big| \tilde{X}_t^{\boldsymbol{\pi}^{\phi}} = x \Big] = 0.
\end{aligned}
$$

Lastly, note

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}} \Big[\int_t^{T \wedge \tau_n} e^{-\beta(s-t)} \check{r}(s, X_s^{\boldsymbol{\pi}^{\phi}}, a_s^{\boldsymbol{\pi}^{\phi}}; \phi) \mathrm{d}s \Big| X_t^{\boldsymbol{\pi}^{\phi}} = x \Big] \\
&= \mathbb{E}^{\mathbb{P}^W} \Big[\int_t^{T \wedge \tau_n} e^{-\beta(s-t)} \int_{\mathcal{A}} \check{r}(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}, a; \phi) \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}^{\phi}} = x \Big].
\end{aligned}
$$

By Assumption 3 and Lemma 2, we get

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}^W} \Big[\int_{\mathcal{A}} |\check{r}(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}, a; \phi)| \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a \Big| \tilde{X}_t^{\boldsymbol{\pi}^{\phi}} = x \Big] \\
\le& C_1 \mathbb{E}^{\mathbb{P}^W} \Big[1 + |\tilde{X}_t^{\boldsymbol{\pi}^{\phi}}|^{\mu} \Big| \tilde{X}_t^{\boldsymbol{\pi}^{\phi}} = x \Big] \le C_2 (1 + |x|^{\mu}).
\end{aligned}
$$

Hence by the dominance convergence theorem, we conclude that as $n \to \infty$,

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}^W} \Big[\int_t^{T \wedge \tau_n} e^{-\beta(s-t)} \int_{\mathcal{A}} \check{r}(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}, a; \phi) \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}^{\phi}} = x \Big] \\
\to& \mathbb{E}^{\mathbb{P}^W} \Big[\int_t^{T} e^{-\beta(s-t)} \int_{\mathcal{A}} \check{r}(s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}, a; \phi) \boldsymbol{\pi}^{\phi}(a|s, \tilde{X}_s^{\boldsymbol{\pi}^{\phi}}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}^{\phi}} = x \Big].
\end{aligned}
$$

This proves the desired result.

Proof of Theorem 6

The proof is similar to the proof of Proposition 4 and that in Jia and Zhou (2021, Proposition 4) by noticing $\boldsymbol{\pi}^*$ satisfies (17).

First, it suffices to consider the case when $\zeta = 0$ because of Theorem 4. For $\eta \in L^2_{\mathcal{F}^{X^{\boldsymbol{\pi}^{\phi^*}}}}\big([0,T], J(\cdot, X^{\boldsymbol{\pi}^{\phi^*}}_{\cdot}; \boldsymbol{\pi}^{\phi^*})\big)$, we write $\eta_s = \boldsymbol{\eta}(s, X^{\boldsymbol{\pi}^{\phi^*}}_{s\wedge\cdot})$. Then the right hand side of (18) can be written as

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}}\Bigg[\int_0^T \boldsymbol{\eta}(s, X^{\boldsymbol{\pi}^{\phi^*}}_{s\wedge\cdot})\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^{\phi^*}(a^{\boldsymbol{\pi}^{\phi^*}}_s|s, X^{\boldsymbol{\pi}^{\phi^*}}_s)\big]\big[\mathrm{d}J(s, X^{\boldsymbol{\pi}}_s; \boldsymbol{\pi}^{\phi^*})\\
&+[r(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a^{\boldsymbol{\pi}^{\phi^*}}_s) + \gamma p\big(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a^{\boldsymbol{\pi}^{\phi^*}}_s, \boldsymbol{\pi}^{\phi^*}(\cdot|s, X^{\boldsymbol{\pi}^{\phi^*}}_s)\big) - \beta J(s, X^{\boldsymbol{\pi}^{\phi^*}}_s; \boldsymbol{\pi}^{\phi^*})]\mathrm{d}s]\\
&+\gamma q(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a^{\boldsymbol{\pi}^{\phi^*}}_s, \phi^*)\mathrm{d}s\Big\}\Big|X^{\boldsymbol{\pi}^{\phi^*}}_0 = x\Bigg]\\
=&\int_0^T \mathbb{E}^{\mathbb{P}}\Bigg[\boldsymbol{\eta}(s, X^{\boldsymbol{\pi}^{\phi^*}}_{s\wedge\cdot})\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^{\phi^*}(a^{\boldsymbol{\pi}^{\phi^*}}_s|s, X^{\boldsymbol{\pi}^{\phi^*}}_s)\big]\big[\mathcal{L}^{a^{\boldsymbol{\pi}^{\phi^*}}_s} J(s, X^{\boldsymbol{\pi}}_s; \boldsymbol{\pi}^{\phi^*})\\
&+r(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a^{\boldsymbol{\pi}^{\phi^*}}_s) + \gamma p\big(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a^{\boldsymbol{\pi}^{\phi^*}}_s, \boldsymbol{\pi}^{\phi^*}(\cdot|s, X^{\boldsymbol{\pi}^{\phi^*}}_s)\big) - \beta J(s, X^{\boldsymbol{\pi}^{\phi^*}}_s; \boldsymbol{\pi}^{\phi^*})\big]\\
&+\gamma q(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a^{\boldsymbol{\pi}^{\phi^*}}_s, \phi^*)\Big\}\Big|X^{\boldsymbol{\pi}^{\phi^*}}_0 = x\Bigg]\mathrm{d}s\\
=&\int_0^T \mathbb{E}^{\mathbb{P}}\Bigg[\boldsymbol{\eta}(s, X^{\boldsymbol{\pi}^{\phi^*}}_{s\wedge\cdot})\int_{\mathcal{A}}\Big\{\big[\frac{\partial}{\partial\phi}\log\boldsymbol{\pi}^{\phi^*}(a|s, X^{\boldsymbol{\pi}^{\phi^*}}_s)\big]\big[\mathcal{L}^{a} J(s, X^{\boldsymbol{\pi}}_s; \boldsymbol{\pi}^{\phi^*})\\
&+r(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a) + \gamma p\big(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a, \boldsymbol{\pi}^{\phi^*}(\cdot|s, X^{\boldsymbol{\pi}^{\phi^*}}_s)\big) - \beta J(s, X^{\boldsymbol{\pi}^{\phi^*}}_s; \boldsymbol{\pi}^{\phi^*})\big]\\
&+\gamma q(s, X^{\boldsymbol{\pi}^{\phi^*}}_s, a, \phi^*)\Big\}\boldsymbol{\pi}^{\phi^*}(a|s, X^{\boldsymbol{\pi}^{\phi^*}}_s)\mathrm{d}a\Big|X^{\boldsymbol{\pi}^{\phi^*}}_0 = x\Bigg]\mathrm{d}s = 0.
\end{aligned}
$$

Proof of Lemma 7

Apply Itô's lemma to $J(\tilde{X}^{\boldsymbol{\pi}}_s; \boldsymbol{\pi})$ on $s \in [t, T]$ to obtain

$$
\begin{aligned}
&\mathbb{E}^{\mathbb{P}^W}\Big[J(\tilde{X}^{\boldsymbol{\pi}}_T; \boldsymbol{\pi})\Big|\tilde{X}^{\boldsymbol{\pi}}_t = x\Big] - J(x; \boldsymbol{\pi})\\
=&\mathbb{E}^{\mathbb{P}^W}\Big[\int_t^T\int_{\mathcal{A}}\mathcal{L}^a J(\tilde{X}^{\boldsymbol{\pi}}_s; \boldsymbol{\pi})\boldsymbol{\pi}(a|\tilde{X}^{\boldsymbol{\pi}}_s)\mathrm{d}a\mathrm{d}s\Big|\tilde{X}^{\boldsymbol{\pi}}_t = x\Big]\\
=&\mathbb{E}^{\mathbb{P}^W}\Big[\int_t^T\int_{\mathcal{A}} -r(\tilde{X}^{\boldsymbol{\pi}}_s, a) - \gamma p\big(\tilde{X}^{\boldsymbol{\pi}}_s, a, \boldsymbol{\pi}(\cdot|\tilde{X}^{\boldsymbol{\pi}}_s)\big)\boldsymbol{\pi}(a|\tilde{X}^{\boldsymbol{\pi}}_s)\mathrm{d}a\mathrm{d}s\Big|\tilde{X}^{\boldsymbol{\pi}}_t = x\Big] + V(\boldsymbol{\pi})(T-t).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\frac{1}{T}\mathbb{E}^{\mathbb{P}}\Big[\int_t^T\big[r(X^{\boldsymbol{\pi}}_s, a^{\boldsymbol{\pi}}_s) + \gamma p\big(X^{\boldsymbol{\pi}}_s, a^{\boldsymbol{\pi}}_s, \boldsymbol{\pi}(\cdot|X^{\boldsymbol{\pi}}_s)\big)\big]\mathrm{d}s\Big|X^{\boldsymbol{\pi}}_t = x\Big]\\
=&\frac{1}{T}\mathbb{E}^{\mathbb{P}^W}\Big[\int_t^T\int_{\mathcal{A}}\big[r(\tilde{X}^{\boldsymbol{\pi}}_s, a) + \gamma p\big(\tilde{X}^{\boldsymbol{\pi}}_s, a, \boldsymbol{\pi}(\cdot|\tilde{X}^{\boldsymbol{\pi}}_s)\big)\big]\boldsymbol{\pi}(a|\tilde{X}^{\boldsymbol{\pi}}_s)\mathrm{d}a\mathrm{d}s\Big|\tilde{X}^{\boldsymbol{\pi}}_t = x\Big]\\
=&V(\boldsymbol{\pi})\frac{T-t}{T} + \frac{1}{T}J(x; \boldsymbol{\pi}) - \frac{1}{T}\mathbb{E}^{\mathbb{P}^W}\Big[J(\tilde{X}^{\boldsymbol{\pi}}_T; \boldsymbol{\pi})\Big|\tilde{X}^{\boldsymbol{\pi}}_t = x\Big].
\end{aligned}
$$

By a similar localization argument as in the proof of Theorem 5, we can show that $\limsup_{T\to\infty}\mathbb{E}^{\mathbb{P}^W}\left[J(\tilde{X}_T^{\boldsymbol{\pi}};\boldsymbol{\pi})\Big|\tilde{X}_t^{\boldsymbol{\pi}}=x\right]$ is finite and independent of x. Taking limit $T\to\infty$ on both sides of the above yields (20).

The above analysis also implies

$$J(\tilde{X}_t^{\boldsymbol{\pi}};\boldsymbol{\pi})+\int_0^t\int_{\mathcal{A}}[r(\tilde{X}_s^{\boldsymbol{\pi}},a)+\gamma p\big(\tilde{X}_s^{\boldsymbol{\pi}},a,\boldsymbol{\pi}(\cdot|\tilde{X}_s^{\boldsymbol{\pi}})\big)-V(\boldsymbol{\pi})]\boldsymbol{\pi}(a|\tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}s$$

is an $(\mathcal{F}^{\tilde{X}^{\boldsymbol{\pi}}},\mathbb{P}^W)$-martingale. For the same reason as in the proof of Theorem 4, we arrive at the second desired conclusion of the lemma.

References

V. Aleksandrov, V. Sysoev, and V. Shemeneva. Stochastic optimization. *Engineering Cybernetics*, 5:11–16, 1968.

A. Arapostathis, V. S. Borkar, and M. K. Ghosh. *Ergodic control of diffusion processes*, volume 143. Cambridge University Press, 2012.

L. C. Baird. Advantage updating. Technical report, Write Lab Wright-Patterson Air Force Base, OH 45433-7301, USA, 1993.

A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, (5):834–846, 1983.

S. Basak and G. Chabakauri. Dynamic mean-variance asset allocation. *The Review of Financial Studies*, 23(8):2970–3016, 2010.

C. Beck, M. Hutzenthaler, and A. Jentzen. On nonlinear Feynman–Kac formulas for viscosity solutions of semilinear parabolic partial differential equations. *Stochastics and Dynamics*, page 2150048, 2021.

A. Bensoussan and J. Frehse. On Bellman equations of ergodic control in R^n. In *Applied Stochastic Analysis*, pages 21–29. Springer, 1992.

S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor–critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

T. Björk, A. Murgoci, and X. Y. Zhou. Mean–variance portfolio optimization with state-dependent risk aversion. *Mathematical Finance*, 24(1):1–24, 2014.

V. S. Borkar and M. K. Ghosh. Ergodic control of multidimensional diffusions I: The existence results. *SIAM Journal on Control and Optimization*, 26(1):112–126, 1988.

V. S. Borkar and M. K. Ghosh. Ergodic control of multidimensional diffusions II: Adaptive control. *Applied Mathematics and Optimization*, 21(1):191–220, 1990.

S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(1):33–57, 1996.

M. Dai, Y. Dong, and Y. Jia. Learning equilibrium mean-variance strategy. *SSRN preprint SSRN:3770818*, 2020.

M. Dai, H. Jin, S. Kou, and Y. Xu. A dynamic mean-variance analysis for log returns. *Management Science*, 67(2):1093–1108, 2021.

T. Degris, P. M. Pilarski, and R. S. Sutton. Model-free reinforcement learning with continuous action in practice. In *2012 American Control Conference (ACC)*, pages 2177–2182. IEEE, 2012.

K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12 (1):219–245, 2000.

Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338. PMLR, 2016.

I. Ekeland and A. Lazrak. Being serious about non-commitment: subgame perfect equilibrium in continuous time. *arXiv preprint math/0604264*, 2006.

W. H. Fleming and H. M. Soner. *Controlled Markov Processes and Viscosity Solutions*, volume 25. Springer Science & Business Media, 2006.

N. Frémaux, H. Sprekeler, and W. Gerstner. Reinforcement learning using a continuous time actor-critic framework with spiking neurons. *PLoS Computational Biology*, 9(4): e1003024, 2013.

X. Gao, Z. Q. Xu, and X. Y. Zhou. State-dependent temperature control for Langevin diffusions. *arXiv preprint arXiv:2011.07456*, 2020.

P. W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

X. Guo, R. Xu, and T. Zariphopoulou. Entropy regularization for mean field games with learning. *arXiv preprint arXiv:2010.00145*, 2020.

T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Y. Jia and X. Y. Zhou. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *arXiv preprint arXiv:2108.06655*, 2021.

I. Karatzas and S. Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer, 2014.

J. Lee and R. S. Sutton. Policy iterations for reinforcement learning problems in continuous time and spaceFundamental theory and methods. *Automatica*, 126:109421, 2021.

D. Li and W.-L. Ng. Optimal dynamic portfolio selection: Multiperiod mean-variance formulation. *Mathematical Finance*, 10(3):387–406, 2000.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

H. R. Maei, C. Szepesvari, S. Bhatnagar, D. Precup, D. Silver, and R. S. Sutton. Convergent temporal-difference learning with arbitrary smooth function approximation. In *NIPS*, pages 1204–1212, 2009.

P. Marbach and J. N. Tsitsiklis. Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control*, 46(2):191–209, 2001.

V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937. PMLR, 2016.

J. Schulman, X. Chen, and P. Abbeel. Equivalence between policy gradients and soft Q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, pages 387–395. PMLR, 2014.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

R. H. Strotz. Myopia and inconsistency in dynamic utility maximization. *The Review of Economic Studies*, 23(3):165–180, 1955.

R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 2018.

R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000a.

R. S. Sutton, S. Singh, and D. McAllester. Comparing policy-gradient algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 2000b.

R. S. Sutton, C. Szepesvári, and H. R. Maei. A convergent o(n) temporal-difference algorithm for off-policy learning with linear function approximation. In *NIPS*, 2008.

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 *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000, 2009.

W. Tang, P. Y. Zhang, and X. Y. Zhou. Exploratory HJB equations and their convergence. *arXiv preprint arXiv:2109.10269*, 2021.

K. G. Vamvoudakis and F. L. Lewis. Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5):878–888, 2010.

H. Wang and X. Y. Zhou. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance*, 30(4):1273–1308, 2020.

H. Wang, T. Zariphopoulou, and X. Y. Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21 (198):1–34, 2020.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

J. Yong and X. Y. Zhou. *Stochastic Controls: Hamiltonian Systems and HJB Equations*. New York, NY: Spinger, 1999.

T. Zhao, H. Hachiya, G. Niu, and M. Sugiyama. Analysis and improvement of policy gradient estimation. In *NIPS*, pages 262–270. Citeseer, 2011.

X. Y. Zhou and D. Li. Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics and Optimization*, 42(1):19–33, 2000.