q-Learning in Continuous Time

Yanwei Jia   JIAYW1993@GMAIL.COM
Department of System Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin, NT, Hong Kong

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 the continuous-time counterpart of Q-learning for reinforcement learning (RL) under the entropy-regularized, exploratory diffusion process formulation introduced by Wang et al. (2020). As the conventional (big) Q-function collapses in continuous time, we consider its first-order approximation and coin the term "(little) q-function". This function is related to the instantaneous advantage rate function as well as the Hamiltonian. We develop a "q-learning" theory around the q-function that is independent of time discretization. Given a stochastic policy, we jointly characterize the associated q-function and value function by martingale conditions of certain stochastic processes, in both on-policy and off-policy settings. We then apply the theory to devise different actor–critic algorithms for solving underlying RL problems, depending on whether or not the density function of the Gibbs measure generated from the q-function can be computed explicitly. One of our algorithms interprets the well-known Q-learning algorithm SARSA, and another recovers a policy gradient (PG) based continuous-time algorithm proposed in Jia and Zhou (2022b). Finally, we conduct simulation experiments to compare the performance of our algorithms with those of PG-based algorithms in Jia and Zhou (2022b) and time-discretized conventional Q-learning algorithms.

Keywords: continuous-time reinforcement learning, policy improvement, q-function, martingale, on-policy and off-policy

1 Introduction

A recent series of papers, Wang et al. (2020) and Jia and Zhou (2022a,b), aim at laying an overarching theoretical foundation for reinforcement learning (RL) in continuous time with continuous state space (diffusion processes) and possibly continuous action space. Specifically, Wang et al. (2020) study how to explore strategically (instead of blindly) by formulating an entropy-regularized, distribution-valued stochastic control problem for diffusion processes. Jia and Zhou (2022a) solve the policy evaluation (PE) problem, namely to learn the value function of a given stochastic policy, by characterizing it as a martingale problem. Jia and Zhou (2022b) investigate the policy gradient (PG) problem, that is, to

determine the gradient of a learned value function with respect to the current policy, and show that PG is mathematically a simpler PE problem and thus solvable by the martingale approach developed in Jia and Zhou (2022a). Combining these theoretical results naturally leads to various online and offline actor–critic (AC) algorithms for general model-free (up to the diffusion dynamics) RL tasks, where one learns value functions and stochastic policies simultaneously and alternatingly. Many of these algorithms recover and/or interpret some well-known existing RL algorithms for Markov decision processes (MDPs) that were often put forward in a heuristic and ad hoc manner.

PG updates a policy along the gradient ascent direction to improve it; so PG is an instance of the general policy improvement (PI) approach. On the other hand, one of the earliest and most popular methods for PI is *Q-learning* (Watkins, 1989; Watkins and Dayan, 1992), whose key ingredient is to learn the *Q-function*, a function of state and action. The learned Q-function is then maximized over actions at each state to achieve improvement of the current policy. The resulting Q-learning algorithms, such as the Q-table, SARSA and DQN (deep Q-network), are widely used for their simplicity and effectiveness in many applications.[1] Most importantly though, compared with PG, one of the key advantages of using Q-learning is that it works both on-policy and off-policy (Sutton and Barto, 2018, Chapter 6).

The Q-function, by definition, is a function of the current state and action, assuming that the agent takes a particular action at the *current time* and follows through a given control policy starting from the *next time step*. Therefore, it is *intrinsically* a notion in *discrete* time; that is why Q-learning has been predominantly studied for discrete-time MDPs. In a continuous-time setting, the Q-function collapses to the value function that is independent of action and hence cannot be used to rank and choose the *current* actions. Indeed, Tallec et al. (2019) opine that "there is no Q-function in continuous time". On the other hand, one may propose discretizing time to obtain a discretized Q-function and then apply the existing Q-learning algorithms. However, Tallec et al. (2019) show experimentally that this approach is very sensitive to time discretization and performs poorly with small time steps. Kim et al. (2021) take a different approach: they include the action as a state variable in the continuous-time system by restricting the action process to be absolutely continuous in time with a bounded growth rate. Thus Q-learning becomes a policy evaluation problem with the state–action pair as the new state variable. However, they consider only deterministic dynamic systems and discretize upfront the continuous-time problem. Crucially, that the action must be absolutely continuous is unpractically restrictive because optimal actions are often only measurable in time and have unbounded variations (such as the bang–bang controls) even for deterministic systems.

As a matter of fact, the aforementioned series of papers, Wang et al. (2020) and Jia and Zhou (2022a,b), are characterized by carrying out all the theoretical analysis within the continuous setting and taking observations at discrete times (for computing the to-

---

1. Q-learning is typically combined with various forms of function approximations. For example, linear function approximations (Melo et al., 2008; Zou et al., 2019), kernel-based nearest neighbor regression (Shah and Xie, 2018), or deep neural networks (Mnih et al., 2015; Fan et al., 2020). Q-learning is not necessarily restricted to finite action space in literature; for example, Q-learning with continuous action spaces is studied in Gu et al. (2016). However, Duan et al. (2016) report and claim that the standard Q-learning algorithm may become less efficient for continuous action spaces.

tal reward over time) only when implementing the algorithms. Some advantages of this approach, compared with discretizing time upfront and then applying existing MDP results, are discussed extensively in Doya (2000); Wang et al. (2020); Jia and Zhou (2022a,b). More importantly, the pure continuous-time approach minimizes or eliminates the impact of time-discretization on learning which, as discussed above, becomes critical especially for continuous-time Q-learning.

Now, if we are to study Q-learning strictly within the continuous-time framework without time-discretization or action restrictions, then the first question is what a proper Q-function should be. When time is discretized, Baird (1993) and Mnih et al. (2016) define the so-called *advantage function* that is the difference in value between taking a particular action versus following a given policy at any state; that is, the difference between the state–action value and the state value. Baird (1993, 1994) and Tallec et al. (2019) notice that such an advantage function can be properly scaled with respect to the size of time discretization whose continuous-time limit exists, called the *(instantaneous) advantage rate function* and can be learned. The advantage updating performs better than conventional Q-learning, as shown numerically in Tallec et al. (2019). However, these papers again consider only deterministic dynamic systems and do not fully develop a theory of this rate function. Here in the present paper, we attack the problem directly from the continuous-time perspective. In particular, we rename the advantage rate function as the (little) *q-function*.[2] This q-function in the finite-time episodic setting is a function of the time–state–action triple under a given policy, analogous to the conventional Q-function. However, it is a continuous-time notion because *it does not depend on time-discretization*. This feature is a vital advantage for learning algorithm design as it avoids the sensitivity with respect to the observation and intervention frequency (the step size of time-discretization).

For the entropy-regularized, exploratory setting for diffusion processes (first formulated in Wang et al. 2020) studied in the paper, the q-function of a given policy turns out to be the *Hamiltonian* that includes the infinitesimal generator of the dynamic system and the instantaneous reward function (see Yong and Zhou 1999 for details), plus the temporal dispersion that consists of the time-derivative of the value function and the depreciation from discounting. The paper aims to develop a comprehensive *q-learning* theory around this q-function and accordingly design alternative RL algorithms other than the PG-based AC algorithms in Jia and Zhou (2022b). There are three main questions. The first is to characterize the q-function. For discrete-time MDPs, the PE method is used to learn both the value function and the Q-function because both functions are characterized by similar Bellman equations. By contrast, in the continuous-time diffusion setting, a Bellman equation is only available for the value function (also known as the Feynman–Kac formula, which yields that the value function solves a second-order linear partial differential equation). The second question is to establish a policy improvement theorem based on the q-function and to design corresponding AC algorithms in a sample/data-driven, model-free fashion. The third question is whether the capability of both on- and off-policy learning, a key advantage of Q-learning, is intact in the continuous-time setting.

---

2. We use the *lower case* letter "*q*" to denote this rate function, much in the same way as the typical use of the lower case letter f to denote a probability density function which is the first-order derivative of the cumulative distribution function usually denoted by the upper case letter F.

We provide complete answers to all these questions. First, given a stochastic policy along with its (learned) value function, the corresponding q-function is characterized by the martingale condition of a certain stochastic process with respect to an enlarged filtration (information field) that includes both the original environmental noise and the action randomization. Alternatively, the value function and the q-function can be *jointly* determined by the martingality of the same process. These results suggest that the martingale perspective for PE and PG in Jia and Zhou (2022a,b) continues to work for q-learning. In particular, we devise several algorithms to learn these functions in the same way as the martingale conditions are employed to generate PE and PG algorithms in Jia and Zhou (2022a,b). Interestingly, a temporal–difference (TD) learning algorithm designed from this perspective recovers and, hence, interprets a version of the well-known SARSA algorithm in the discrete-time Q-learning literature. Moreover, inspired by these continuous-time martingale characterizations, in Appendix A we present martingale conditions for Q-learning of *discrete-time* MDPs, which are new to our best knowledge. This demonstrates that the continuous-time RL study may offer new perspectives even in discrete time. Second, we prove that a Gibbs sampler with a properly scaled current q-function (independent of time discretization) as the exponent of its density function improves upon the current policy. This PI result translates into a policy-updating algorithm when the normalizing constant of the Gibbs measure can be explicitly computed. Otherwise, if the normalizing constant is unavailable, we prove a stronger PI theorem in terms of the Kullback–Leibler (KL) divergence, analogous to a result for discrete-time MDPs in Haarnoja et al. (2018a). One of the algorithms out of this theorem happens to recover a PG-based AC algorithm in Jia and Zhou (2022b). Finally, we prove that the aforementioned martingale conditions hold for both the on-policy and off-policy settings. More precisely, the value function and the q-function associated with a given *target* policy can be learned based on a dataset generated by either the target policy itself or an arbitrary *behavior* policy. So, q-learning indeed works off-policy as well.

The rest of the paper is organized as follows. In Section 2 we review Wang et al. (2020)'s entropy-regularized, exploratory formulation for RL in continuous time and space, and present some useful preliminary results. In Section 3, we establish the q-learning theory, including the motivation and definition of the q-function and its on-/off-policy martingale characterizations. q-learning algorithms are developed, depending on whether or not the normalizing constant of the Gibbs measure is available, in Section 4 and Section 5, respectively. Extension to ergodic tasks is presented in Section 6. In Section 7, we illustrate the proposed algorithms and compare them with PG-based algorithms and time-discretized, conventional Q-learning algorithms on two specific examples with simulated data.[3] Finally, Section 8 concludes. The Appendix contains various supplementary materials and proofs of statements in the main text.

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). We use $\mathbb{S}_{++}^k$ to

---

3. The code to reproduce our simulation studies is publicly available at `https://www.dropbox.com/sh/34cgnupnuaix15l/AAAj2yQYfNCOtPUc1_7VhbkIa?dl=0`.

denote all the $k \times k$ symmetric and positive definite matrices. Given two matrices A and B of the same size, denote by $A \circ B$ their inner product, by $|A|$ the Euclidean/Frobenius norm of A, by $\det A$ the determinant when A is a square matrix, and by $A^\top$ the transpose of any matrix A. For $A \in \mathbb{S}_{++}^k$, we write $\sqrt{A} = UD^{1/2}U^\top$, where $A = UDU^\top$ is its eigenvalue decomposition with U an orthogonal matrix, D a diagonal matrix, and $D^{1/2}$ 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. We denote by $\mathcal{N}(\mu, \Sigma)$ the probability density function of the multivariate normal distribution with mean vector μ and covariance matrix Σ. Finally, 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.

2.1 Classical model-based formulation

For readers' convenience, we first recall the classical, model-based stochastic control formulation.

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 $\mathcal{A} \subset \mathbb{R}^m$ is the action/control space. 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 $\left(\Omega, \mathcal{F}, \mathbb{P}^W; \{\mathcal{F}_s^W\}_{s \geq 0}\right)$ along with a standard n-dimensional Brownian motion $W = \{W_s, s \geq 0\}$:

$$dX_s^{\boldsymbol{a}} = b(s, X_s^{\boldsymbol{a}}, \boldsymbol{a}_s)ds + \sigma(s, X_s^{\boldsymbol{a}}, \boldsymbol{a}_s)dW_s, \ s \in [0, T], \tag{1}$$

where $\boldsymbol{a}_s$ stands for the agent's action at time s.

The goal of a stochastic control problem is, for each initial time–state pair $(t, x) \in [0, T) \times \mathbb{R}^d$ of (1), to find the optimal $\{\mathcal{F}_s^W\}_{s \geq 0}$-progressively measurable (continuous-time) sequence of actions $\boldsymbol{a} = \{\boldsymbol{a}_s, t \leq s \leq T\}$ – also called the optimal control or optimal strategy – that maximizes the expected total discounted reward:

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

where r is an (instantaneous) reward function (i.e., the expected rate of reward conditioned on time, state, and action), h is the lump-sum reward function applied at the end of the planning period T, and $\beta \geq 0$ is a constant discount factor that measures the time-value of the payoff or the impatience level of the agent.

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

The (generalized) *Hamiltonian* is a function $H : [0, T] \times \mathbb{R}^d \times \mathcal{A} \times \mathbb{R}^d \times \mathbb{R}^{d \times d} \to \mathbb{R}$ associated with problem (1)–(2) defined as (Yong and Zhou, 1999):

$$H(t, x, a, p, q) = b(t, x, a) \circ p + \frac{1}{2}\sigma\sigma^\top(t, x, a) \circ q + r(t, x, a). \tag{3}$$

We make the same assumptions as in Jia and Zhou (2022b) 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* $\varphi \in \{b, \ \sigma\}$, *there exists a constant* $C > 0$ *such that*

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

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

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

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

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

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, *assuming* that the functional forms of b, σ, r, h are all given and known. A centerpiece of the theory is the Hamilton–Jacobi–Bellman (HJB) equation

$$\frac{\partial V^*}{\partial t}(t, x) + \sup_{a \in \mathcal{A}} H\left(t, x, a, \frac{\partial V^*}{\partial x}(t, x), \frac{\partial^2 V^*}{\partial x^2}(t, x)\right) - \beta V^*(t, x) = 0, \ V^*(T, x) = h(x). \quad (4)$$

Here, $\frac{\partial V^*}{\partial x} \in \mathbb{R}^d$ is the gradient and $\frac{\partial^2 V^*}{\partial x^2} \in \mathbb{R}^{d \times d}$ is the Hessian matrix.

Under proper conditions, the unique solution (possibly in the sense of viscosity solution) to (4) is the optimal value function to the problem (1)–(2), i.e.,

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

Moreover, the following function, which maps a time–state pair to an action:

$$a^*(t, x) = \arg\sup_{a \in \mathcal{A}} H\left(t, x, a, \frac{\partial V^*}{\partial x}(t, x), \frac{\partial^2 V^*}{\partial x^2}(t, x)\right) \quad (5)$$

is the optimal (non-stationary, feedback) control *policy* of the problem. In view of the definition of the Hamiltonian (3), the maximization in (5) indicates that, at any given time and state, one ought to maximize a suitably weighted average of the (myopic) instantaneous reward and the risk-adjusted, (long-term) positive impact on the system dynamics. See Yong and Zhou (1999) for a detailed account of this theory and many discussions on its economic interpretations and implications.

2.2 Exploratory formulation in reinforcement learning

We now present the RL formulation of the problem to be studied in this paper. In the RL setting, the agent has partial or simply no knowledge about the environment (i.e. the functions b, σ, r, h). What she can do is "trial and error" – to try a (continuous-time) sequence of actions $\boldsymbol{a} = \{a_s, t \le s \le T\}$, observe the corresponding state process $X^{\boldsymbol{a}} = \{X_s^{\boldsymbol{a}}, t \le s \le T\}$ along with both a stream of discounted running rewards $\{e^{-\beta(s-t)}r(s, X_s^{\boldsymbol{a}}, a_s), t \le s \le T\}$ and a discounted, end-of-period lump-sum reward $e^{-\beta(T-t)}h(X_T^{\boldsymbol{a}})$ where β is a given, known discount factor, and continuously update and improve her actions based on these observations.[4]

A critical question is how to strategically *generate* these trial-and-error sequences of actions. The idea is *randomization*; namely, the agent devises and employs a *stochastic policy*, which is a probability distribution on the action space, to generate actions according to the current time–state pair. It is important to note that such randomization itself is *independent* of the underlying Brownian motion W, the random source of the original control problem that stands for the environmental noises. Specifically, assume the probability space is rich enough to support uniformly distributed random variables on $[0, 1]$ that is independent of W, and then such a uniform random variable can be used to generate other random variables with density functions. Let $\{Z_t, 0 \le t \le T\}$ be a process of mutually independent copies of a uniform random variable on $[0, 1]$, the construction of which requires a suitable extension of probability space; see Sun (2006) for details. We then further expand the filtered probability space to $(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_s\}_{s \ge 0})$ where $\mathcal{F}_s = \mathcal{F}_s^W \vee \sigma(Z_t, 0 \le t \le s)$ and the probability measure $\mathbb{P}$, now defined on $\mathcal{F}_T$, is an extension from $\mathbb{P}^W$ (i.e. the two probability measures coincide when restricted to $\mathcal{F}_T^W$).

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.[5] At each time s, an action a_s is generated or sampled from the distribution $\boldsymbol{\pi}(\cdot|s, X_s)$.

Fix a stochastic policy $\boldsymbol{\pi}$ and an initial time–state pair (t, x). Consider the following SDE

$$\mathrm{d}X_s^{\boldsymbol{\pi}} = b(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}})\mathrm{d}s + \sigma(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}})\mathrm{d}W_s, \ \ s \in [t, T]; \ \ X_t^{\boldsymbol{\pi}} = x \tag{6}$$

defined on $(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_s\}_{s \ge 0})$, where $a^{\boldsymbol{\pi}} = \{a_s^{\boldsymbol{\pi}}, t \le s \le T\}$ is an $\{\mathcal{F}_s\}_{s \ge 0}$-progressively measurable action process generated from $\boldsymbol{\pi}$. (6) is an SDE with random coefficients, whose well-posedness (i.e. existence and uniqueness of solution) is established in Yong and Zhou (1999, Chapter 1, Theorem 6.16) under Assumption 1. Fix $a^{\boldsymbol{\pi}}$, the unique solution to (6), $X^{\boldsymbol{\pi}} = \{X_s^{\boldsymbol{\pi}}, t \le s \le T\}$, is the sample state process corresponding to $a^{\boldsymbol{\pi}}$ that solves (1).[6] Moreover, following Wang et al. (2020), we add an entropy regularizer to the reward

---

4. This procedure applies to both the offline and online settings. In the former, the agent can repeatedly try different sequences of actions over the full time period $[0, T]$, record the corresponding state processes and payoffs, and then learn and update the policy based on the resulting dataset. In the latter, the agent updates the actions as she goes, based on all the up-to-date historical observations.

5. 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.

6. Here, $X^{\boldsymbol{\pi}}$ also depends on the *specific* copy $a^{\boldsymbol{\pi}}$ sampled from $\boldsymbol{\pi}$; however, to ease notation we write $X^{\boldsymbol{\pi}}$ instead of $X^{\boldsymbol{\pi}, a^{\boldsymbol{\pi}}}$.

function to encourage exploration (represented by the stochastic policy), leading to

$$
\begin{aligned}
J(t, x; \boldsymbol{\pi}) =& \mathbb{E}^{\mathbb{P}} \bigg[\int_t^T e^{-\beta(s-t)} \left[r(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_s^{\boldsymbol{\pi}} | s, X_s^{\boldsymbol{\pi}}) \right] \mathrm{d}s \\
&+ e^{-\beta(T-t)} h(X_T^{\boldsymbol{\pi}}) \Big| X_t^{\boldsymbol{\pi}} = x \bigg],
\end{aligned}
\tag{7}
$$

where $\mathbb{E}^{\mathbb{P}}$ is the expectation with respect to both the Brownian motion and the action randomization, and $\gamma \geq 0$ is a given weighting parameter on exploration also known as the *temperature* parameter.

Wang et al. (2020) consider the following SDE

$$
\mathrm{d}X_s = \tilde{b}\big(s, X_s, \boldsymbol{\pi}(\cdot | s, X_s)\big) \mathrm{d}t + \tilde{\sigma}\big(s, X_s, \boldsymbol{\pi}(\cdot | s, X_s)\big) \mathrm{d}W_s, \ s \in [t, T]; \quad X_t = x, \tag{8}
$$

where

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

Intuitively, based on the law of large number, the solution of (8), denoted by $\{\tilde{X}_s^{\boldsymbol{\pi}}, t \leq s \leq T\}$, is the limit of the average of the sample trajectories $X^{\boldsymbol{\pi}}$ over randomization (i.e., copies of $\boldsymbol{\pi}$). Rigorously, it follows from the property of Markovian projection due to Brunick and Shreve (2013, Theorem 3.6) that $X_s^{\boldsymbol{\pi}}$ and $\tilde{X}_s^{\boldsymbol{\pi}}$ have the same distribution for each $s \in [t, T]$. Hence, the value function (7) is identical to

$$
J(t, x; \boldsymbol{\pi}) = \mathbb{E}^{\mathbb{P}^W} \bigg[\int_t^T e^{-\beta(s-t)} \tilde{\mathcal{R}}(s, \tilde{X}_s^{\boldsymbol{\pi}}, \boldsymbol{\pi}(\cdot | s, \tilde{X}_s^{\boldsymbol{\pi}})) \mathrm{d}s + e^{-\beta(T-t)} h(\tilde{X}_T^{\boldsymbol{\pi}}) \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x \bigg]. \tag{9}
$$

where $\tilde{\mathcal{R}}(s, x, \pi) = \int_{\mathcal{A}} [r(s, x, a) - \gamma \log \pi(a)] \pi(a) \mathrm{d}a$.

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

$$
J^*(t, x) = \max_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} J(t, x; \boldsymbol{\pi}), \tag{10}
$$

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

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

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

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

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

(iii) For any given $\alpha > 0$, the entropy of $\boldsymbol{\pi}$ and its α-moment have polynomial growth in x, i.e., there are constants $C = C(\alpha) > 0$ and $\mu' = \mu'(\alpha) \geq 1$ such that $|\int_{\mathcal{A}} -\log \boldsymbol{\pi}(a|t,x)\boldsymbol{\pi}(a|t,x)\mathrm{d}a| \leq C(1 + |x|^{\mu'})$, and $\int_{\mathcal{A}} |a|^\alpha \boldsymbol{\pi}(a|t,x)\mathrm{d}a \leq C(1 + |x|^{\mu'}) \; \forall(t,x)$.

Under Assumption 1 along with (i) and (ii) in Definition 1, Jia and Zhou (2022b, Lemma 2) show that the SDE (8) admits a unique strong solution since its coefficients are locally Lipschitz continuous and have linear growth; see Mao (2007, Chapter 2, Theorem 3.4) for the general result. In addition, the growth condition (iii) guarantees that the new payoff function (9) is finite. More discussions on the conditions required in the above definition can be found in Jia and Zhou (2022b). Moreover, Definition 1 only contains the conditions on the policy, hence they can be verified.

Note that the problem (8)–(9) is *mathematically* equivalent to the problem (6)–(7). Nevertheless, 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 for algorithm design.

2.3 Some useful preliminary results

Lemma 2 in Jia and Zhou (2022b) yields that the value function of a given admissible policy $\boldsymbol{\pi} = \boldsymbol{\pi}(\cdot|\cdot,\cdot) \in \boldsymbol{\Pi}$ satisfies the following PDE

$$
\begin{aligned}
&\frac{\partial J}{\partial t}(t,x;\boldsymbol{\pi}) + \int_{\mathcal{A}} \left[H\left(t,x,a,\frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}),\frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\right) - \gamma \log \boldsymbol{\pi}(a|t,x) \right] \boldsymbol{\pi}(a|t,x)\mathrm{d}a \\
&- \beta J(t,x;\boldsymbol{\pi}) = 0, \\
&J(T,x;\boldsymbol{\pi}) = h(x).
\end{aligned}
\tag{11}
$$

This is a version of the celebrated Feynman–Kac formula in the current RL setting. Under Assumption 1 as well as the admissibility conditions in Definition 1, the PDE (11) admits a unique viscosity solution among polynomially growing functions (Beck et al., 2021, Theorem 1.1).

On the other hand, Tang et al. (2022, Section 5) derive the following *exploratory HJB equation* for the problem (8)–(9) satisfied by the optimal value function:

$$
\frac{\partial J^*}{\partial t}(t,x) + \sup_{\boldsymbol{\pi} \in \mathcal{P}(\mathcal{A})} \int_{\mathcal{A}} \left[H\left(t,x,a,\frac{\partial J^*}{\partial x}(t,x),\frac{\partial^2 J^*}{\partial x^2}(t,x)\right) - \gamma \log \boldsymbol{\pi}(a) \right] \boldsymbol{\pi}(a)\mathrm{d}a - \beta J^*(t,x;\boldsymbol{\pi}) = 0,
$$
$$
J^*(T,x) = h(x).
\tag{12}
$$

Moreover, the optimal (stochastic) policy is a *Gibbs measure* or *Boltzmann distribution*:

$$
\boldsymbol{\pi}^*(\cdot|t,x) \propto \exp\left\{ \frac{1}{\gamma} H\left(t,x,\cdot,\frac{\partial J^*}{\partial x}(t,x),\frac{\partial^2 J^*}{\partial x^2}(t,x)\right) \right\}
$$

or, after normalization,

$$
\boldsymbol{\pi}^*(a|t,x) = \frac{\exp\{\frac{1}{\gamma} H\left(t,x,a,\frac{\partial J^*}{\partial x}(t,x),\frac{\partial^2 J^*}{\partial x^2}(t,x)\right)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma} H\left(t,x,a,\frac{\partial J^*}{\partial x}(t,x),\frac{\partial^2 J^*}{\partial x^2}(t,x)\right)\}\mathrm{d}a}.
\tag{13}
$$

This result shows that one should use a Gibbs sampler in general to generate trial-and-error strategies to explore the environment when the regularizer is chosen to be entropy.[7]

In the special case when the system dynamics are linear in action and payoffs are quadratic in action, the Hamiltonian is a quadratic function of action and Gibbs thus specializes to Gaussian under mild conditions.

Plugging (13) to (12) to replace the supremum operator therein leads to the following equivalent form of the exploratory HJB equation

$$
\frac{\partial J^*}{\partial t}(t,x) + \gamma \log \left[\int_{\mathcal{A}} \exp \left\{ \frac{1}{\gamma} H \left(t, x, a, \frac{\partial J^*}{\partial x}(t,x), \frac{\partial^2 J^*}{\partial x^2}(t,x) \right) \right\} \mathrm{d}a \right] - \beta J^*(t,x) = 0,
$$
$$
J^*(T,x) = h(x). \tag{14}
$$

More theoretical properties regarding (14) and consequently the problem (6)–(7) including its well-posedness can be found in Tang et al. (2022).

The Gibbs measure can also be used to derive a *policy improvement theorem*. Wang and Zhou (2020) prove such a theorem in the context of continuous-time mean–variance portfolio selection, which is essentially a linear–quadratic (LQ) control problem. The following extends that result to the general case.

Theorem 2 *For any given $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, define $\boldsymbol{\pi}'(\cdot|t,x) \propto \exp\{\frac{1}{\gamma} H\big(t,x,\cdot,\frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\}$. If $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$, then*

$$
J(t,x;\boldsymbol{\pi}') \geq J(t,x;\boldsymbol{\pi}).
$$

Moreover, if the following map

$$
\mathcal{I}(\boldsymbol{\pi}) = \frac{\exp\{\frac{1}{\gamma} H\big(t,x,\cdot,\frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma} H\big(t,x,a,\frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\} \mathrm{d}a}, \quad \boldsymbol{\pi} \in \boldsymbol{\Pi}
$$

has a fixed point $\boldsymbol{\pi}^*$, then $\boldsymbol{\pi}^*$ is the optimal policy.

At this point, Theorem 2 remains a *theoretical* result that cannot be directly applied to learning procedures, because the Hamiltonian $H\big(t,x,a,\frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)$ depends on the knowledge of the model parameters which we do not have in the RL context. To develop implementable algorithms, we turn to Q-learning.

3 q-Function in Continuous Time: The Theory

This section is the theoretical foundation of the paper, with the analysis entirely in continuous time. We start with defining a Q-function parameterized by a time step $\Delta t > 0$, and then motivate the notion of q-function that is independent of Δt. We further provide martingale characterizations of the q-function.

---

7. We stress here that the Gibbs sampler is due to the entropy-regularizer, and it is possible to have other types of distribution for exploration. For example, Han et al. (2022) propose a family of regularizers that lead to exponential, uniform, and ε-greedy exploratory policies. O'Donoghue (2021); Gao et al. (2020) study how to choose time and state dependent temperature parameters. One may also carry out exploration via randomizing value function instead of policies; see e.g. Osband et al. (2019).

3.1 Q-function

Tallec et al. (2019) consider a continuous-time MDP and then discretize it upfront to a discrete-time MDP with time discretization δt. In that setting, the authors argue that "there is no Q-function in continuous time" because the Q-function collapses to the value function when δt is infinitesimally small. Here, we take an entirely different approach that does not involve time discretization. Incidentally, by comparing the form of the Gibbs measure (13) with that of the widely employed Boltzmann exploration for learning in discrete-time MDPs, Gao et al. (2020) and Zhou (2021) *conjecture* that the continuous-time counterpart of the Q-function is the Hamiltonian. Our approach will also provide a rigorous justification on and, indeed, a refinement of this conjecture.

Given $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ and $(t, x, a) \in [0, T) \times \mathbb{R}^d \times \mathcal{A}$, consider a "perturbed" policy of $\boldsymbol{\pi}$, denoted by $\tilde{\boldsymbol{\pi}}$, as follows: It takes the action $a \in \mathcal{A}$ on $[t, t + \Delta t]$ where $\Delta t > 0$, and then follows $\boldsymbol{\pi}$ on $[t + \Delta t, T]$. The corresponding state process $X^{\tilde{\boldsymbol{\pi}}}$, given $X_t^{\tilde{\boldsymbol{\pi}}} = x$, can be broken into two pieces. On $[t, t + \Delta t)$, it is X^a which is the solution to

$$\mathrm{d}X_s^a = b(s, X_s^a, a)\mathrm{d}s + \sigma(s, X_s^a, a)\mathrm{d}W_s, \ s \in [t, t + \Delta t); X_t^a = x,$$

while on $[t + \Delta t, T]$, it is $X^{\boldsymbol{\pi}}$ following (6) but with the initial time–state pair $(t + \Delta t, X_{t+\Delta t}^a)$.

With $\Delta t > 0$ fixed, we introduce the (Δt-parameterized) Q-function, denoted by $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$, to be the expected reward drawn from the perturbed policy, $\tilde{\boldsymbol{\pi}}$:[8]

$$
\begin{aligned}
&Q_{\Delta t}(t, x, a; \boldsymbol{\pi}) \\
=&\mathbb{E}^{\mathbb{P}}\bigg[\int_t^{t+\Delta t} e^{-\beta(s-t)} r(s, X_s^a, a)\mathrm{d}s \\
&+ \int_{t+\Delta t}^T e^{-\beta(s-t)} [r(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_s^{\boldsymbol{\pi}} | s, X_s^{\boldsymbol{\pi}})]\mathrm{d}s + e^{-\beta(T-t)} h(X_T^{\boldsymbol{\pi}}) \Big| X_t^{\tilde{\boldsymbol{\pi}}} = x \bigg].
\end{aligned}
$$

Recall that our formulation (7) includes an entropy regularizer term that incentivizes exploration using stochastic policies. However, in defining $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$ above we do not include such a term on $[t, t + \Delta t)$ for $\tilde{\boldsymbol{\pi}}$ because a deterministic constant action a is applied whose entropy is excluded.

The following proposition provides an expansion of this Q-function in Δt.

Proposition 3 *We have*

$$
\begin{aligned}
&Q_{\Delta t}(t, x, a; \boldsymbol{\pi}) \\
=&J(t, x; \boldsymbol{\pi}) + \left[\frac{\partial J}{\partial t}(t, x; \boldsymbol{\pi}) + H\big(t, x, a, \frac{\partial J}{\partial x}(t, x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t, x; \boldsymbol{\pi})\big) - \beta J(t, x; \boldsymbol{\pi}) \right] \Delta t + o(\Delta t).
\end{aligned}
$$
(15)

So, this Δt-based Q-function includes three terms. The leading term is the value function J, which is *independent* of the action a taken. This is natural because we only apply a for a time window of length Δt and hence the impact of this action is only of the order Δt.

---

8. In the existing literature, Q-learning with entropy regularization is often referred to as the "soft Q-learning" with the associated "soft Q-function". A review of and more discussions on soft Q-learning in discrete time can be found in Appendix A as well as in Haarnoja et al. (2018b); Schulman et al. (2017).

Next, the first-order term of Δt is the Hamiltonian plus the temporal change of the value function (consisting of the derivative of the value function in time and the discount term, both of which would disappear for stationary and non-discounted problems), in which *only* the Hamiltonian depends on the action a. It is interesting to note that these are the same terms in the Feynman–Kac PDE (11), less the entropy term. Finally, the higher-order residual term, $o(\Delta t)$, comes from the approximation error of the integral over $[t, t+\Delta t]$ and can be ignored when such errors are aggregated as Δt gets smaller.

Proposition 3 yields that this Q-function also collapses to the value function when $\Delta t \to 0$, similar to what Tallec et al. (2019) claim. However, we must emphasize that our Q-function is *not* the same as the one used in Tallec et al. (2019), who discretize time *upfront* and study the resulting discrete-time MDP. The corresponding Q-function and value function therein exist based on the discrete-time RL theory, denoted respectively by $Q^{\delta t}, V^{\delta t}$ where δt is the time discretization size. Then Tallec et al. (2019) argue that the advantage function $A^{\delta t} = Q^{\delta t} - V^{\delta t}$ can be properly scaled by δt, leading to the existence of $\lim_{\delta t \to 0} \frac{A^{\delta t}}{\delta t}$. In short, the Q-function and the resulting algorithms in Tallec et al. (2019) are still in the realm of discrete-time MDPs. By contrast, $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$ introduced here is a *continuous-time* notion. It is the value function of a policy that applies a constant action in $[t, t + \Delta t]$ and follows $\boldsymbol{\pi}$ thereafter. So Δt here is a *parameter* representing the length of period in which the constant action is taken, *not* the time discretization size as in Tallec et al. (2019). Most importantly, our Q-function is just a tool used to introduce the q-function, the centerpiece of this paper. Having said all these, we recognize that $\lim_{\delta t \to 0} \frac{A^{\delta t}}{\delta t}$ as identified by Tallec et al. (2019) is the counterpart of our q-function in their setting, and that we arrive at this same object in two different ways (discretizing upfront then taking $\delta t \to 0$, versus a true continuous-time analysis).

3.2 q-function

Since the leading term in the Q-function, $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$, coincides with the value function of $\boldsymbol{\pi}$ and hence can not be used to rank action a, we focus on the first-order approximation, which gives an infinitesimal state–action value. Motivated by this, we define the "*q-function*" as follows:

Definition 4 *The q-function of the problem* (6)–(7) *associated with a given policy* $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ *is defined as*

$$q(t, x, a; \boldsymbol{\pi}) = \frac{\partial J}{\partial t}(t, x; \boldsymbol{\pi}) + H\left(t, x, a, \frac{\partial J}{\partial x}(t, x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t, x; \boldsymbol{\pi})\right) - \beta J(t, x; \boldsymbol{\pi}),$$
$$(t, x, a) \in [0, T] \times \mathbb{R}^d \times \mathcal{A}. \tag{16}$$

Clearly, this function is the first-order *derivative* of the Q-function with respect to Δt, as an immediate consequence of Proposition 3:

Corollary 5 *We have*

$$q(t, x, a; \boldsymbol{\pi}) = \lim_{\Delta t \to 0} \frac{Q_{\Delta t}(t, x, a; \boldsymbol{\pi}) - J(t, x; \boldsymbol{\pi})}{\Delta t}. \tag{17}$$

Some remarks are in order. First, the q-function is a function of the time–state–action triple under a given policy, analogous to the conventional Q-function for MDPs; yet it is a continuous-time notion because *it does not depend on any time-discretization.* This feature is a vital advantage for learning algorithm design as Tallec et al. (2019) point out that the performance of RL algorithms is very sensitive with respect to the time discretization. Second, the q-function is related to the advantage function in the MDP literature (e.g., Baird 1993; Mnih et al. 2016), which describes the difference between the state–action value and the state value. Here in this paper, the q-function reflects the *instantaneous advantage rate* of a given action at a given time–state pair under a given policy.

We also notice that in (16) only the Hamiltonian depends on a; hence the improved policy presented in Theorem 2 can also be expressed in terms of the q-function:

$$\boldsymbol{\pi}'(\cdot|t,x) \propto \exp\left\{\frac{1}{\gamma}H\big(t,x,\cdot,\frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}),\frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\right\} \propto \exp\left\{\frac{1}{\gamma}q(t,x,\cdot;\boldsymbol{\pi})\right\}.$$

Consequently, if we can learn the q-function $q(\cdot,\cdot,\cdot;\boldsymbol{\pi})$ under any policy $\boldsymbol{\pi}$, then it follows from Theorem 2 that we can improve $\boldsymbol{\pi}$ by implementing a Gibbs measure over the q-values, analogous to, say, ε-greedy policy in classical Q-learning (Sutton and Barto, 2018, Chapter 6). Finally, the analysis so far justifies the aforementioned conjecture about the proper continuous-time version of the Q-function, and provides a theoretical interpretation to the widely used Boltzmann exploration in RL.

The main theoretical results of this paper are martingale characterizations of the q-function, which can in turn be employed to devise algorithms for learning crucial functions including the q-function, in the same way as in applying the martingale approach for PE (Jia and Zhou, 2022a).[9]

The following first result characterizes the q-function associated with a given policy $\boldsymbol{\pi}$, assuming that its value function has already been learned and known.

Theorem 6 *Let a policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, its value function J and a continuous function $\hat{q}: [0,T] \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ be given. Then*

(i) *$\hat{q}(t,x,a) = q(t,x,a;\boldsymbol{\pi})$ for all $(t,x,a) \in [0,T] \times \mathbb{R}^d \times \mathcal{A}$ if and only if for all $(t,x) \in [0,T] \times \mathbb{R}^d$, the following process*

$$e^{-\beta s}J(s,X_s^{\boldsymbol{\pi}};\boldsymbol{\pi}) + \int_t^s e^{-\beta u}[r(u,X_u^{\boldsymbol{\pi}},a_u^{\boldsymbol{\pi}}) - \hat{q}(u,X_u^{\boldsymbol{\pi}},a_u^{\boldsymbol{\pi}})]\mathrm{d}u \qquad (18)$$

is an $(\{\mathcal{F}_s\}_{s\geq 0}, \mathbb{P})$-martingale, where $\{X_s^{\boldsymbol{\pi}}, t \leq s \leq T\}$ is the solution to (6) under $\boldsymbol{\pi}$ with $X_t^{\boldsymbol{\pi}} = x$.

(ii) *If $\hat{q}(t,x,a) = q(t,x,a;\boldsymbol{\pi})$ for all $(t,x,a) \in [0,T] \times \mathbb{R}^d \times \mathcal{A}$, then for any $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$ and for all $(t,x) \in [0,T] \times \mathbb{R}^d$, the following process*

$$e^{-\beta s}J(s,X_s^{\boldsymbol{\pi}'};\boldsymbol{\pi}) + \int_t^s e^{-\beta u}[r(u,X_u^{\boldsymbol{\pi}'},a_u^{\boldsymbol{\pi}'}) - \hat{q}(u,X_u^{\boldsymbol{\pi}'},a_u^{\boldsymbol{\pi}'})]\mathrm{d}u \qquad (19)$$

is an $(\{\mathcal{F}_s\}_{s\geq 0}, \mathbb{P})$-martingale, where $\{X_s^{\boldsymbol{\pi}'}, t \leq s \leq T\}$ is the solution to (6) under $\boldsymbol{\pi}'$ with initial condition $X_t^{\boldsymbol{\pi}'} = x$.

---

9. Some discrete-time counterparts of these results are presented in Appendix A.

(iii) If there exists $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$ such that for all $(t, x) \in [0, T] \times \mathbb{R}^d$, (19) is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale where $X_t^{\boldsymbol{\pi}'} = x$, then $\hat{q}(t, x, a) = q(t, x, a; \boldsymbol{\pi})$ for all $(t, x, a) \in [0, T] \times \mathbb{R}^d \times \mathcal{A}$.

Moreover, in any of the three cases above, the q-function satisfies

$$\int_{\mathcal{A}} \left[q(t, x, a; \boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}(a|t, x) \right] \boldsymbol{\pi}(a|t, x) \mathrm{d}a = 0, \quad \forall (t, x) \in [0, T] \times \mathbb{R}^d. \tag{20}$$

Jia and Zhou (2022a) unify and characterize the learning of value function (i.e., PE) by martingale conditions of certain stochastic processes. Theorem 6 shows that learning the q-function again boils down to maintaining the martingality of the processes (18) or (19). However, there are subtle differences between these martingale conditions. Jia and Zhou (2022a) consider only deterministic policies so the martingales therein are with respect to $(\{\mathcal{F}_s^W\}_{s \geq 0}, \mathbb{P}^W)$. Jia and Zhou (2022b) extends the policy evaluation to include stochastic policies but the related martingales are in terms of the *averaged* state $\tilde{X}^{\boldsymbol{\pi}}$ and hence also with respect to $(\{\mathcal{F}_s^W\}_{s \geq 0}, \mathbb{P}^W)$. By contrast, the martingality in Theorem 6 is with respect to $(\{\mathcal{F}_t\}_{t \geq 0}, \mathbb{P})$, where the filtration $\{\mathcal{F}_t\}_{t \geq 0}$ is the *enlarged* one that includes the randomness for generating actions. Note that the filtration determines the class of test functions necessary to characterize a martingale through the so-called martingale orthogonality conditions (Jia and Zhou, 2022a). So the above martingale conditions suggest that one ought to choose test functions dependent of the past and current actions when designing algorithms to learn the q-function. Moreover, the q-function can be interpreted as a compensator to guarantee the martingality over this larger information field.[10]

Theorem 6-(i) informs *on-policy* learning, namely, learning the q-function of the given policy $\boldsymbol{\pi}$, called the *target policy*, based on data $\{(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}), t \leq s \leq T\}$ generated by $\boldsymbol{\pi}$ itself. Nevertheless, one of the key advantages of classical Q-learning for MDPs is that it also works for *off-policy* learning, namely, learning the q-function of a given target policy $\boldsymbol{\pi}$ based on data generated by a different admissible policy $\boldsymbol{\pi}'$, called a *behavior policy*. On-policy and off-policy reflect two different learning settings depending on the availability of data and/or the choice of a learning agent. When data generation can be controlled by the agent, conducting on-policy learning is possible although she could still elect off-policy learning. By contrast, when data generation is not controlled by the agent and she has to rely on data under other policies, then it becomes off-policy learning. Theorem 6-(ii) and -(iii) stipulate that our q-learning in continuous time can also be off-policy.

Finally, (20) is a consistency condition to uniquely identify the q-function. If we only consider deterministic policies of the form $a(\cdot, \cdot)$ in which case $\gamma = 0$ and $\boldsymbol{\pi}(\cdot|t, x)$ degenerates into the Dirac measure concentrating on $a(t, x)$, then (20) reduces to

$$q(t, x, a(t, x); a) = 0,$$

which corresponds to equation (23) in Tallec et al. (2019).

The following result strengthens Theorem 6, in the sense that it characterizes the value function and the q-function under a given policy *simultaneously*.

---

10. More precisely, $\int_0^t e^{-\beta s} q(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}; \boldsymbol{\pi}) \mathrm{d}s$ is the compensator of $e^{-\beta t} J(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + \int_0^t e^{-\beta s} r(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) \mathrm{d}s$. Recall that a compensator of an adapted stochastic process Y_t is a predictable process A_t such that $Y_t - A_t$ is a local martingale. Intuitively speaking, the compensator is the drift part of a diffusion process, extracting the trend of the process.

Theorem 7 *Let a policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, a function $\hat{J} \in C^{1,2}\big([0,T] \times \mathbb{R}^d\big) \cap C\big([0,T] \times \mathbb{R}^d\big)$ with polynomial growth, and a continuous function $\hat{q} : [0,T] \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ be given satisfying*

$$\hat{J}(T,x) = h(x), \quad \int_{\mathcal{A}} \big[\hat{q}(t,x,a) - \gamma \log \boldsymbol{\pi}(a|t,x)\big] \boldsymbol{\pi}(a|t,x) \mathrm{d}a = 0, \quad \forall (t,x) \in [0,T] \times \mathbb{R}^d. \quad (21)$$

Then

(i) *$\hat{J}$ and $\hat{q}$ are respectively the value function and the q-function associated with $\boldsymbol{\pi}$ if and only if for all $(t,x) \in [0,T] \times \mathbb{R}^d$, the following process*

$$e^{-\beta s} \hat{J}(s, X_s^{\boldsymbol{\pi}}) + \int_t^s e^{-\beta u} [r(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}) - \hat{q}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})] \mathrm{d}u \quad (22)$$

is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale, where $\{X_s^{\boldsymbol{\pi}}, t \leq s \leq T\}$ is the solution to (6) with $X_t^{\boldsymbol{\pi}} = x$.

(ii) *If $\hat{J}$ and $\hat{q}$ are respectively the value function and the q-function associated with $\boldsymbol{\pi}$, then for any $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$ and all $(t,x) \in [0,T] \times \mathbb{R}^d$, the following process*

$$e^{-\beta s} \hat{J}(s, X_s^{\boldsymbol{\pi}'}) + \int_t^s e^{-\beta u} [r(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'}) - \hat{q}(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'})] \mathrm{d}u \quad (23)$$

is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale, where $\{X_s^{\boldsymbol{\pi}'}, t \leq s \leq T\}$ is the solution to (6) under $\boldsymbol{\pi}'$ with initial condition $X_t^{\boldsymbol{\pi}'} = x$.

(iii) *If there exists $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$ such that for all $(t,x) \in [0,T] \times \mathbb{R}^d$, (23) is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale where $X_t^{\boldsymbol{\pi}'} = x$, then $\hat{J}$ and $\hat{q}$ are respectively the value function and the q-function associated with $\boldsymbol{\pi}$.*

Moreover, in any of the three cases above, if it holds further that $\boldsymbol{\pi}(a|t,x) = \frac{\exp\{\frac{1}{\gamma}\hat{q}(t,x,a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}\hat{q}(t,x,a)\}\mathrm{d}a}$, then $\boldsymbol{\pi}$ is the optimal policy and $\hat{J}$ is the optimal value function.

Theorem 7 characterizes the value function and the q-function in terms of a *single* martingale condition, in each of the on-policy and off-policy settings, which will be the foundation for designing learning algorithms in this paper. The conditions in (21) ensure that $\hat{J}$ corresponds to the correct terminal payoff function and $\hat{q}$ corresponds to the soft q-function with the entropy regularizer. In particular, if the policy is the Gibbs measure generated by $\frac{1}{\gamma}\hat{q}$, then $\hat{J}$ and $\hat{q}$ are respectively the value function and the q-function under the optimal policy. Finally, note the (subtle) difference between Theorem 7 and Theorem 6. There may be multiple $(\hat{J}, \hat{q})$ pairs satisfying the martingale conditions of (22) or (23) in Theorem 7; so the conditions in (21) are *required* for identifying the correct value function and q-function.[11] By contrast, these conditions are *implied* if the correct value function has already been known and given, as in Theorem 6.

---

11. If the terminal condition $\hat{J}(T,x) = h(x)$ is changed to $\hat{J}(T,x) = \hat{h}(x)$, then $(\hat{J}, \hat{q})$ satisfying the martingale condition (22) or (23) would be the value function and q-function corresponding to a different learning task with the pair of running and terminal reward functions being $(r, \hat{h})$. If the normalization condition on $\hat{q}$ is missing, say $\int_{\mathcal{A}} \big[\hat{q}(t,x,a) - \gamma \log \boldsymbol{\pi}(a|t,x)\big] \boldsymbol{\pi}(a|t,x) \mathrm{d}a = \hat{r}(t,x)$ holds instead, then $(\hat{J}, \hat{q})$ would be the value function and q-function corresponding to the learning task with the running and terminal reward functions $(r - \hat{r}, h)$.

3.3 Optimal q-function

We now focus on the q-function associated with the *optimal* policy $\boldsymbol{\pi}^*$ in (13). Based on Definition 4, it is defined as

$$q^*(t, x, a) = \frac{\partial J^*}{\partial t}(t, x) + H\left(t, x, a, \frac{\partial J^*}{\partial x}(t, x), \frac{\partial^2 J^*}{\partial x^2}(t, x)\right) - \beta J^*(t, x), \qquad (24)$$

where J^* is the optimal value function satisfying the exploratory HJB equation (14).

Proposition 8 *We have*

$$\int_{\mathcal{A}} \exp\{\frac{1}{\gamma} q^*(t, x, a)\} \mathrm{d}a = 1, \qquad (25)$$

for all (t, x), and consequently the optimal policy $\boldsymbol{\pi}^*$ is

$$\boldsymbol{\pi}^*(a|t, x) = \exp\{\frac{1}{\gamma} q^*(t, x, a)\}. \qquad (26)$$

As seen from its proof (in Appendix C), Proposition 8 is an immediate consequence of the exploratory HJB equation (14). Indeed, (25) is equivalent to (14) when viewed as an equation for J^* due to (24). However, in terms of q^*, satisfying (25) is only a *necessary* condition or a minimum requirement for being the optimal q-function, and by no means sufficient, in the current RL setting. This is because in the absence of knowledge about the primitives b, σ, r, h, we are unable to determine J^* from q^* by their relationship (24) which can be viewed as a PDE for J^*. To fully characterize J^* as well as q^*, we will have to resort to martingale condition, as stipulated in the following result.

Theorem 9 *Let a function $\widehat{J^*} \in C^{1,2}\big([0, T] \times \mathbb{R}^d\big) \cap C\big([0, T] \times \mathbb{R}^d\big)$ with polynomial growth and a continuous function $\widehat{q^*} : [0, T] \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ be given satisfying*

$$\widehat{J^*}(T, x) = h(x), \quad \int_{\mathcal{A}} \exp\{\frac{1}{\gamma}\widehat{q^*}(t, x, a)\} \mathrm{d}a = 1, \quad \forall (t, x) \in [0, T] \times \mathbb{R}^d. \qquad (27)$$

Then

(i) *If $\widehat{J^*}$ and $\widehat{q^*}$ are respectively the optimal value function and the optimal q-function, then for any $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ and all $(t, x) \in [0, T] \times \mathbb{R}^d$, the following process*

$$e^{-\beta s}\widehat{J^*}(s, X_s^{\boldsymbol{\pi}}) + \int_t^s e^{-\beta u}[r(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}) - \widehat{q^*}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})]\mathrm{d}u \qquad (28)$$

is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale, where $\{X_s^{\boldsymbol{\pi}}, t \leq s \leq T\}$ is the solution to (6) under $\boldsymbol{\pi}$ with $X_t^{\boldsymbol{\pi}} = x$. Moreover, in this case, $\widehat{\boldsymbol{\pi}^*}(a|t, x) = \exp\{\frac{1}{\gamma}\widehat{q^*}(t, x, a)\}$ is the optimal policy.

(ii) *If there exists one $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ such that for all $(t, x) \in [0, T] \times \mathbb{R}^d$, (28) is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale, then $\widehat{J^*}$ and $\widehat{q^*}$ are respectively the optimal value function and the optimal q-function.*

Theorem 9 lays a foundation for off-policy learning without having to go through iterative policy improvement. We emphasize here that to learn the optimal policy and the optimal q-function, it is infeasible to conduct on-policy learning because $\boldsymbol{\pi}^*$ is unknown and hence the constraints (21) can not be checked nor enforced. The new constraints in (27) no longer depend on policies and instead give basic requirements for the candidate value function and q-function. Maintaining the martingality of (28) under any admissible policy is equivalent to the optimality of the candidate value function and q-function. Moreover, Theorem 9-(ii) suggests that we do not need to use data generated by *all* admissible policies. Instead, those generated by *one* – any one – policy should suffice; e.g. we could take $\boldsymbol{\pi}(a|t,x) = \exp\{\frac{1}{\gamma}\widehat{q^*}(t,x,a)\}$ as the behavior policy.

Finally, let us remark that Theorems 7 and 9 are independent of each other in the sense that one does not imply the other. One may design different algorithms from them depending on different cases and needs.

4 q-Learning Algorithms When Normalizing Constant Is Available

This section and the next discuss algorithm design based on the theoretical results in the previous section. Theorems 7 and 9 provide the theoretical foundation for designing both on-policy and off-policy algorithms to *simultaneously* learn and update the value function (critic) and the policy (actor) with proper function approximations of J and q. In this section, we present these actor–critic, q-learning algorithms when the normalizing constant involved in the Gibbs measure is available or computable.[12]

4.1 q-Learning algorithms

Given a policy $\boldsymbol{\pi}$, to learn its associated value function and q-function, we denote by J^θ and q^ψ the parameterized function approximators that satisfy the two constraints:

$$J^\theta(T,x) = h(x), \ \int_{\mathcal{A}} \left[q^\psi(t,x,a) - \gamma \log \boldsymbol{\pi}(a|t,x)\right] \boldsymbol{\pi}(a|t,x)\mathrm{d}a = 0, \qquad (29)$$

for all $\theta \in \Theta \subset \mathbb{R}^{L_\theta}, \psi \in \Psi \subset \mathbb{R}^{L_\psi}$. Then Theorem 2 suggests that the policy can be improved by

$$\boldsymbol{\pi}^\psi(a|t,x) = \frac{\exp\{\frac{1}{\gamma}q^\psi(t,x,a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}q^\psi(t,x,a)\}\mathrm{d}a}, \qquad (30)$$

while by assumption the normalizing constant $\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}q^\psi(t,x,a)\}\mathrm{d}a$ for any $\psi \in \Psi$ can be explicitly computed. Next, we can use the data generated by $\boldsymbol{\pi}^\psi$ and apply the martingale condition in Theorem 7 to learn its value function and q-function, leading to an iterative actor–critic algorithm.

---

12. Conventional Q-learning or SARSA is often said to be value-based in which the Q-function is a state–action value function serving as a critic. In q-learning, as Theorem 6 stipulates, the q-function is uniquely determined by the value function J as its compensator. Hence, q plays a dual role here: it is both a critic (as it can be derived endogenously from the value function) and an actor (as it derives an improved policy). This is why we call the q-learning algorithms actor–critic, if with a slight abuse of the term because the "actor" here is not purely exogenous as with, say, policy gradient.

Alternatively, we may choose to directly learn the optimal value function and q-function based on Theorem 9. In this case, the approximators J^θ and q^ψ should satisfy

$$J^\theta(T,x) = h(x), \quad \int_{\mathcal{A}} \exp\{\frac{1}{\gamma} q^\psi(t,x,a)\} \mathrm{d}a = 1. \tag{31}$$

Note that if the policy $\boldsymbol{\pi}$ in (29) is taken in the form (30), then the second equation in (31) is automatically satisfied. Henceforth in this section, we focus on deriving algorithms based on Theorem 9 and hence always impose the constraints (31). In this case, any approximator of the q-function directly gives that of the policy via $\boldsymbol{\pi}^\psi(a|t,x) = \exp\{\frac{1}{\gamma} q^\psi(t,x,a)\}$; thereby we avoid learning the q-function and the policy separately. Moreover, making use of (31) typically results in more special parametric form of the q-function approximator q^ψ, potentially facilitating more efficient learning. Here is an example.

Example 1 When the system dynamic is linear in action a and reward is quadratic in a, the Hamiltonian, and hence the q-function, is quadratic in a. So we can parameterize

$$q^\psi(t,x,a) = -\frac{1}{2} q_2^\psi(t,x) \circ \left(a - q_1^\psi(t,x)\right) \left(a - q_1^\psi(t,x)\right)^\top + q_0^\psi(t,x), \quad (t,x,a) \in [0,T] \times \mathbb{R}^d \times \mathbb{R}^m,$$

with $q_1^\psi(t,x) \in \mathbb{R}^m$, $q_2^\psi(t,x) \in \mathbb{S}_{++}^m$ and $q_0^\psi(t,x) \in \mathbb{R}$. The corresponding policy is a multivariate normal distribution for which the normalizing constant can be computed:

$$\boldsymbol{\pi}^\psi(\cdot|t,x) = \mathcal{N}\left(q_1^\psi(t,x), \gamma\left(q_2^\psi(t,x)\right)^{-1}\right),$$

with its entropy value being $-\frac{1}{2}\log\det q_2^\psi(t,x) + \frac{m}{2}\log 2\pi e\gamma$. The second constraint on q^ψ in (31) then yields $q_0^\psi(t,x) = \frac{\gamma}{2}\log\left(\det q_2^\psi(t,x)\right) - \frac{m\gamma}{2}\log 2\pi$. This in turn gives rise to a more specific parametric form

$$q^\psi(t,x,a) = -\frac{1}{2} q_2^\psi(t,x) \circ \left(a - q_1^\psi(t,x)\right) \left(a - q_1^\psi(t,x)\right)^\top + \frac{\gamma}{2}\log\left(\det q_2^\psi(t,x)\right) - \frac{m\gamma}{2}\log 2\pi.$$

The next step in algorithm design is to update (θ,ψ) by enforcing the martingale condition stipulated in Theorem 9 and applying the techniques developed in Jia and Zhou (2022a). A number of algorithms can be developed based on two types of objectives: to minimize the martingale loss function or to satisfy the martingale orthogonality conditions. The latter calls for solving a system of equations for which there are further two different techniques: applying stochastic approximation as in the classical temporal–difference (TD) algorithms or minimizing a quadratic function in lieu of the system of equations as in the generalized methods of moment (GMM). For reader's convenience, we summarize these methods in the q-learning context below.

- Minimize the martingale loss function:

$$\frac{1}{2}\mathbb{E}^\mathbb{P}\left[\int_0^T \left[e^{-\beta(T-t)} h(X_T^{\boldsymbol{\pi}^\psi}) - J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + \int_t^T e^{-\beta(s-t)}[r(s, X_s^{\boldsymbol{\pi}^\psi}, a_s^{\boldsymbol{\pi}^\psi}) - q^\psi(s, X_s^{\boldsymbol{\pi}^\psi}, a_s^{\boldsymbol{\pi}^\psi})]\mathrm{d}s\right]^2 \mathrm{d}t\right].$$

This method is intrinsically offline because the loss function involves the whole horizon $[0, T]$; however we are free to choose optimization algorithms to update the parameters (θ, ψ). For example, we can apply stochastic gradient decent to update

$$\theta \leftarrow \theta + \alpha_\theta \int_0^T \frac{\partial J^\theta}{\partial \theta}(t, X_t^{\boldsymbol{\pi}^\psi}) G_{t:T} \mathrm{d}t$$

$$\psi \leftarrow \psi + \alpha_\psi \int_0^T \int_t^T e^{-\beta(s-t)} \frac{\partial q^\psi}{\partial \psi}(s, X_s^{\boldsymbol{\pi}^\psi}, a_s^{\boldsymbol{\pi}^\psi}) \mathrm{d}s G_{t:T} \mathrm{d}t,$$

where $G_{t:T} = e^{-\beta(T-t)} h(X_T^{\boldsymbol{\pi}^\psi}) - J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + \int_t^T e^{-\beta(s-t)} [r(s, X_s^{\boldsymbol{\pi}^\psi}, a_s^{\boldsymbol{\pi}^\psi}) - q^\psi(s, X_s^{\boldsymbol{\pi}^\psi}, a_s^{\boldsymbol{\pi}^\psi})] \mathrm{d}s$, and α_θ and α_ψ are the learning rates. We present Algorithm 1 based on this updating rule. Note that this algorithm is analogous to the classical gradient Monte Carlo method or TD(1) for MDPs (Sutton and Barto, 2018) because full sample trajectories are used to compute gradients.

- Choose two different test functions ξ_t and ζ_t that are both $\mathcal{F}_t$-adapted vector-valued stochastic processes, and consider the following system of equations in (θ, ψ):

$$\mathbb{E}^{\mathbb{P}} \left[\int_0^T \xi_t \left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t \right] \right] = 0,$$

and

$$\mathbb{E}^{\mathbb{P}} \left[\int_0^T \zeta_t \left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t \right] \right] = 0.$$

To solve these equations iteratively, we use stochastic approximation to update (θ, ψ) either offline by[13]

$$\theta \leftarrow \theta + \alpha_\theta \int_0^T \xi_t \left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t \right],$$

$$\psi \leftarrow \psi + \alpha_\psi \int_0^T \zeta_t \left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t \right],$$

or online by

$$\theta \leftarrow \theta + \alpha_\theta \xi_t \left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t \right],$$

$$\psi \leftarrow \psi + \alpha_\psi \zeta_t \left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) \mathrm{d}t \right].$$

Typical choices of test functions are $\xi_t = \frac{\partial J^\theta}{\partial \theta}(t, X_t^{\boldsymbol{\pi}^\psi})$, $\zeta_t = \frac{\partial q^\psi}{\partial \psi}(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})$ yielding algorithms that would be closest to TD-style policy evaluation and Q-learning (e.g., in Tallec et al. 2019) and at the same time belong to the more general semi-gradient methods (Sutton and Barto, 2018). We present the online and offline q-learning algorithms, Algorithms 2 and 3 respectively, based on these test functions. We must, however,

---

13. Here, when implementing in a computer program, $\mathrm{d}J^\theta$ is the timestep-wise difference in J^θ when time is discretized, or indeed it is the temporal-difference (TD) of the value function approximator J^θ.

stress that testing against these two specific functions is theoretically *not* sufficient to guarantee the martingale condition. Using them with a rich, large-dimensional parametric family for J and q makes approximation to the martingale condition finer and finer. Moreover, the corresponding stochastic approximation algorithms are not guaranteed to converge in general, and the test functions have to be carefully selected (Jia and Zhou, 2022a) depending on (J, q). Some of the different test functions leading to different types of algorithms in the context of policy evaluation are discussed in Jia and Zhou (2022a).

- Choose the same types of test functions ξ_t and ζ_t as above but now minimize the GMM objective functions:

$$\mathbb{E}^{\mathbb{P}}\left[\int_0^T \xi_t\left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t\right]\right]^\top$$
$$A_\theta \mathbb{E}^{\mathbb{P}}\left[\int_0^T \xi_t\left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t\right]\right],$$

and

$$\mathbb{E}^{\mathbb{P}}\left[\int_0^T \zeta_t\left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t\right]\right]^\top$$
$$A_\psi \mathbb{E}^{\mathbb{P}}\left[\int_0^T \zeta_t\left[\mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - q^\psi(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t\right]\right],$$

  where $A_\theta \in \mathbb{S}_{++}^{L_\theta}, A_\psi \in \mathbb{S}_{++}^{L_\psi}$. Typical choices of these matrices are $A_\theta = I_{L_\theta}$ and $A_\psi = I_{L_\psi}$, or $A_\theta = (\mathbb{E}^{\mathbb{P}}[\int_0^T \xi_t \xi_t^\top \mathrm{d}t])^{-1}$ and $A_\psi = (\mathbb{E}^{\mathbb{P}}[\int_0^T \zeta_t \zeta_t^\top \mathrm{d}t])^{-1}$. Again, we refer the reader to Jia and Zhou (2022a) for discussions on these choices and the connection with the classical GTD algorithms and GMM method.

4.2 Connections with SARSA

With a fixed time-step size Δt, we can define our Q-function, $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$, by (15), parameterize it by $Q_{\Delta t}^\varphi(t, x, a; \boldsymbol{\pi})$, and then apply existing (big) Q-learning algorithms such as SARSA (cf., Sutton and Barto 2018, Chapter 10) to learn the parameter φ. Now, how do we compare this approach of Δt-based Q-learning with that of q-learning?

Equation (15) suggests that $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$ can be approximated by

$$Q_{\Delta t}(t, x, a; \boldsymbol{\pi}) \approx J(t, x; \boldsymbol{\pi}) + q(t, x, a; \boldsymbol{\pi})\Delta t.$$

Our q-learning method is to learn separately the zeroth-order and first-order terms of $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$ in Δt, and the two terms are in themselves *independent* of Δt. Our approach therefore has a true "continuous-time" nature without having to rely on Δt or any time disretization, which theoretically facilitates the analysis carried out in the previous section and algorithmically mitigates the high sensitivity with respect to time discretization. Our approach is similar to that introduced in Baird (1994) and Tallec et al. (2019) where the value function and the (rescaled) advantage function are approximated separately.

Algorithm 1 Offline–Episodic q-Learning ML Algorithm

Inputs: initial state x_0, horizon T, time step Δt, number of episodes N, number of mesh grids K, initial learning rates $\alpha_\theta, \alpha_\psi$ and a learning rate schedule function $l(\cdot)$ (a function of the number of episodes), functional forms of parameterized value function $J^\theta(\cdot, \cdot)$ and q-function $q^\psi(\cdot, \cdot, \cdot)$ satisfying (31), and temperature parameter γ.

Required program (on-policy): 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 instantaneous reward r at time t as outputs. Policy $\boldsymbol{\pi}^\psi(a|t, x) = \exp\{\frac{1}{\gamma} q^\psi(t, x, a)\}$.

Required program (off-policy): observations $\{a_{t_k}, r_{t_k}, x_{t_{k+1}}\}_{k=0, \cdots, K-1}$ $\cup$ $\{x_{t_K}, h(x_{t_K})\} = Observation(\Delta t)$ including the observed actions, rewards, and state trajectories under the given behavior policy at the sampling time grids with step size Δt.

Learning procedure:

  Initialize θ, ψ.

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

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

    {**On-policy case**

    while $k < K$ **do**

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

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

      Update $k \leftarrow k + 1$.

    end while

    }

    {**Off-policy case** Obtain one observation $\{a_{t_k}, r_{t_k}, x_{t_{k+1}}\}_{k=0, \cdots, K-1} \cup \{x_{t_K}, h(x_{t_K})\} = Observation(\Delta t)$.

    }

    For every $k = 0, 1, \cdots, K - 1$, compute

$$G_{t_k:T} = e^{-\beta(T-t_k)} h(x_{t_K}) - J^\theta(t_k, x_{t_k}) + \sum_{i=k}^{K-1} e^{-\beta(t_i - t_k)} [r_{t_i} - q^\psi(t_i, x_{t_i}, a_{t_i})] \Delta t.$$

    Update θ and ψ by

$$\theta \leftarrow \theta + l(j)\alpha_\theta \sum_{k=0}^{K-1} \frac{\partial J^\theta}{\partial \theta}(t_k, x_{t_k}) G_{t_k:T} \Delta t.$$

$$\psi \leftarrow \psi + l(j)\alpha_\psi \sum_{k=0}^{K-1} \left[\sum_{i=k}^{K-1} \frac{\partial q^\psi}{\partial \psi}(t_k, x_{t_k}, a_{t_k}) \Delta t \right] G_{t_k:T} \Delta t.$$

  end for

Algorithm 2 Offline–Episodic q-Learning Algorithm

Inputs: initial state x_0, horizon T, time step Δt, number of episodes N, number of mesh grids K, initial learning rates $\alpha_\theta, \alpha_\psi$ and a learning rate schedule function $l(\cdot)$ (a function of the number of episodes), functional forms of parameterized value function $J^\theta(\cdot, \cdot)$ and q-function $q^\psi(\cdot, \cdot, \cdot)$ satisfying (31), functional forms of test functions $\boldsymbol{\xi}(t, x_{\cdot \wedge t}, a_{\cdot \wedge t})$ and $\boldsymbol{\zeta}(t, x_{\cdot \wedge t}, a_{\cdot \wedge t})$, and temperature parameter γ.

Required program (on-policy): 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 instantaneous reward r at time t as outputs. Policy $\boldsymbol{\pi}^\psi(a|t, x) = \exp\{\frac{1}{\gamma} q^\psi(t, x, a)\}$.

Required program (off-policy): observations $\{a_{t_k}, r_{t_k}, x_{t_{k+1}}\}_{k=0,\cdots,K-1} \cup \{x_{t_K}, h(x_{t_K})\} = Observation(\Delta t)$ including the observed actions, rewards, and state trajectories under the given behavior policy at the sampling time grids with step size Δt.

Learning procedure:

  Initialize θ, ψ.

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

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

    {**On-policy case**

    while $k < K$ **do**

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

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

      Update $k \leftarrow k + 1$.

    end while

    }

    {**Off-policy case** Obtain one observation $\{a_{t_k}, r_{t_k}, x_{t_{k+1}}\}_{k=0,\cdots,K-1} \cup \{x_{t_K}, h(x_{t_K})\} = Observation(\Delta t)$.

    }

    For every $k = 0, 1, \cdots, K - 1$, compute and store test functions $\xi_{t_k} = \boldsymbol{\xi}(t_k, x_{t_0}, \cdots, x_{t_k}, a_{t_0}, \cdots, a_{t_k})$, $\zeta_{t_k} = \boldsymbol{\zeta}(t_k, x_{t_0}, \cdots, x_{t_k}, a_{t_0}, \cdots, a_{t_k})$.

    Compute

$$\Delta\theta = \sum_{i=0}^{K-1} \xi_{t_i} \big[J^\theta(t_{i+1}, x_{t_{i+1}}) - J^\theta(t_i, x_{t_i}) + r_{t_i}\Delta t - q^\psi(t_i, x_{t_i}, a_{t_i})\Delta t - \beta J^\theta(t_i, x_{t_i})\Delta t \big],$$

$$\Delta\psi = \sum_{i=0}^{K-1} \zeta_{t_i} \big[J^\theta(t_{i+1}, x_{t_{i+1}}) - J^\theta(t_i, x_{t_i}) + r_{t_i}\Delta t - q^\psi(t_i, x_{t_i}, a_{t_i})\Delta t - \beta J^\theta(t_i, x_{t_i})\Delta t \big].$$

    Update θ and ψ by

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

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

  end for

Algorithm 3 Online-Incremental q-Learning Algorithm

Inputs: initial state x_0, horizon T, time step Δt, number of mesh grids K, initial learning rates $\alpha_\theta, \alpha_\psi$ and learning rate schedule function $l(\cdot)$ (a function of the number of episodes), functional forms of parameterized value function $J^\theta(\cdot, \cdot)$ and q-function $q^\psi(\cdot, \cdot, \cdot)$ satisfying (31), functional forms of test functions $\boldsymbol{\xi}(t, x_{\cdot \wedge t}, a_{\cdot \wedge t})$ and $\boldsymbol{\zeta}(t, x_{\cdot \wedge t}, a_{\cdot \wedge t})$, and temperature parameter γ.

Required program (on-policy): 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 instantaneous reward r at time t as outputs. Policy $\boldsymbol{\pi}^\psi(a|t, x) = \exp\{\frac{1}{\gamma} q^\psi(t, x, a)\}$.

Required program (off-policy): observations $\{a, r, x'\} = Observation(t, x; \Delta t)$ including the observed actions, rewards, and state when the current time-state pair is (t, x) under the given behavior policy at the sampling time grids with step size Δt.

Learning procedure:

   Initialize θ, ψ.

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

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

      while $k < K$ **do**

         {**On-policy case**

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

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

         }

         {**Off-policy case**

         Obtain one observation $a_{t_k}, r_{t_k}, x_{t_{k+1}} = Observation(t_k, x_{t_k}; \Delta t)$.

         }

         Compute test functions $\xi_{t_k} = \boldsymbol{\xi}(t_k, x_{t_0}, \cdots, x_{t_k}, a_{t_0}, \cdots, a_{t_k})$, $\zeta_{t_k} = \boldsymbol{\zeta}(t_k, x_{t_0}, \cdots, x_{t_k}, a_{t_0}, \cdots, a_{t_k})$.

         Compute

$$\delta = J^\theta(t_{k+1}, x_{t_{k+1}}) - J^\theta(t_k, x_{t_k}) + r_{t_k}\Delta t - q^\psi(t_k, x_{t_k}, a_{t_k})\Delta t - \beta J^\theta(t_k, x_{t_k})\Delta t,$$
$$\Delta\theta = \xi_{t_k}\delta,$$
$$\Delta\psi = \zeta_{t_k}\delta.$$

         Update θ and ψ by
$$\theta \leftarrow \theta + l(j)\alpha_\theta\Delta\theta.$$
$$\psi \leftarrow \psi + l(j)\alpha_\psi\Delta\psi.$$

         Update $k \leftarrow k + 1$

      end while

   end for

However, as pointed out already, as much as the two approaches may lead to certain similar algorithms, they are different conceptually. The ones in Baird (1994) and Tallec et al. (2019) are still based on discrete-time MDPs and hence depend on the size of time-discretization. By contrast, the value function and q-function in q-learning are well-defined quantities in continuous time and independent of any time-discretization.

On the other hand, approximating the value function by J^θ and the q-function by q^ψ separately yields a specific approximator of the Q-function by

$$Q_{\Delta t}^{\theta,\psi}(t,x,a) \approx J^\theta(t,x) + q^\psi(t,x,a)\Delta t. \tag{32}$$

With this Q-function approximator $Q^{\theta,\psi}$, one of our q-learning based algorithms actually recovers (a modification of) SARSA, one of the most well-known conventional Q-learning algorithm.

To see this, consider state X_t at time t. Then the conventional SARSA with the approximator $Q^{\theta,\psi}$ updates parameters (θ, ψ) by

$$\theta \leftarrow \theta + \alpha_\theta \bigg(Q_{\Delta t}^{\theta,\psi}(t+\Delta t, X_{t+\Delta t}^{a_t^{\pi^\psi}}, a_{t+\Delta t}^{\pi^\psi}) - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}^{\pi^\psi}|t+\Delta t, X_{t+\Delta t}^{a_t^{\pi^\psi}})\Delta t$$

$$- Q_{\Delta t}^{\theta,\psi}(t, X_t, a_t^{\pi^\psi}) + r(t, X_t, a_t^{\pi^\psi})\Delta t - \beta Q_{\Delta t}^{\theta,\psi}(t, X_t, a_t^{\pi^\psi})\Delta t \bigg) \frac{\partial Q_{\Delta t}^{\theta,\psi}}{\partial \theta}(t, X_t, a_t^{\pi^\psi}),$$

$$\psi \leftarrow \psi + \alpha_\psi \bigg(Q_{\Delta t}^{\theta,\psi}(t+\Delta t, X_{t+\Delta t}^{a_t^{\pi^\psi}}, a_{t+\Delta t}^{\pi^\psi}) - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}^{\pi^\psi}|t+\Delta t, X_{t+\Delta t}^{a_t^{\pi^\psi}})\Delta t$$

$$- Q_{\Delta t}^{\theta,\psi}(t, X_t, a_t^{\pi^\psi}) + r(t, X_t, a_t^{\pi^\psi})\Delta t - \beta Q_{\Delta t}^{\theta,\psi}(t, X_t, a_t^{\pi^\psi})\Delta t \bigg) \frac{\partial Q_{\Delta t}^{\theta,\psi}}{\partial \psi}(t, X_t, a_t^{\pi^\psi}),$$

$$\tag{33}$$

where $a_t^{\pi^\psi} \sim \boldsymbol{\pi}^\psi(\cdot|t, X_t)$ and $a_{t+\Delta t}^{\pi^\psi} \sim \boldsymbol{\pi}^\psi(\cdot|t+\Delta t, X_{t+\Delta t}^{a_t})$.

Note that by (32), we have

$$\frac{\partial Q_{\Delta t}^{\theta,\psi}}{\partial \theta} \approx \frac{\partial J_{\Delta t}^\theta}{\partial \theta}, \quad \frac{\partial Q_{\Delta t}^{\theta,\psi}}{\partial \psi} \approx \frac{\partial q_{\Delta t}^\psi}{\partial \psi}\Delta t.$$

This means that $\frac{\partial Q_{\Delta t}^{\theta,\psi}}{\partial \psi}$ has an additional order of Δt when compared with $\frac{\partial Q_{\Delta t}^{\theta,\psi}}{\partial \theta}$, resulting in a much slower rate of update on ψ than θ. This observation is also made by Tallec et al. (2019), who suggest as a remedy modifying the update rule for Q-function by replacing the partial derivative of the Q-function with that of the advantage function. In our setting, this means we change $\frac{\partial Q_{\Delta t}^{\theta,\psi}}{\partial \psi}$ to $\frac{\partial q_{\Delta t}^\psi}{\partial \psi}$ in the second update rule of (33).

Now, the terms in the brackets in (33) can be further rewritten as:

$$
\begin{aligned}
& Q_{\Delta t}^{\theta,\psi}(t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}}, a_{t+\Delta t}^{\boldsymbol{\pi}^\psi}) - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}^{\boldsymbol{\pi}^\psi}|t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}})\Delta t \\
& \quad - Q_{\Delta t}^{\theta,\psi}(t, X_t, a_t^{\boldsymbol{\pi}^\psi}) + r(t, X_t, a_t^{\boldsymbol{\pi}^\psi})\Delta t - \beta Q_{\Delta t}^{\theta,\psi}(t, X_t, a_t^{\boldsymbol{\pi}^\psi})\Delta t \\
=& J^\theta(t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}}) - J^\theta(t, X_t) + [q^\psi(t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}}, a_{t+\Delta t}^{\boldsymbol{\pi}^\psi}) - q^\psi(t, X_t, a_t^{\boldsymbol{\pi}^\psi})]\Delta t \\
& \quad + r(t, X_t, a_t^{\boldsymbol{\pi}^\psi})\Delta t - \beta J^\theta(t, X_t)\Delta t - \beta q^\psi(t, X_t, a_t^{\boldsymbol{\pi}^\psi})(\Delta t)^2 - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}^{\boldsymbol{\pi}^\psi}|t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}})\Delta t \\
=& J^\theta(t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}}) - J^\theta(t, X_t) - q^\psi(t, X_t, a_t^{\boldsymbol{\pi}^\psi})\Delta t + r(t, X_t, a_t^{\boldsymbol{\pi}^\psi})\Delta t - \beta J^\theta(t, X_t)\Delta t \\
& \quad + \left[q^\psi(t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}}, a_{t+\Delta t}^{\boldsymbol{\pi}^\psi}) - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}^{\boldsymbol{\pi}^\psi}|t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}}) \right] \Delta t - \beta q^\psi(t, X_t, a_t^{\boldsymbol{\pi}^\psi})(\Delta t)^2 \\
\approx& \mathrm{d}J^\theta(t, X_t) - q^\psi(t, X_t, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t + r(t, X_t, a_t^{\boldsymbol{\pi}^\psi})\mathrm{d}t - \beta J^\theta(t, X_t)\mathrm{d}t \\
& \quad + \underbrace{\left[q^\psi(t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}}, a_{t+\Delta t}^{\boldsymbol{\pi}^\psi}) - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}^{\boldsymbol{\pi}^\psi}|t+\Delta t, X_{t+\Delta t}^{a_t^{\boldsymbol{\pi}^\psi}}) \right]}_{\text{mean-0 term due to (29)}} \mathrm{d}t,
\end{aligned}
$$

where in the last step we drop the higher-order small term $(\Delta t)^2$ while replacing the difference terms with differential ones.

Comparing (33) with the updating rule in Algorithm 3 using test functions $\xi_t = \frac{\partial J^\theta}{\partial \theta}(t, X_t^{\boldsymbol{\pi}^\psi})$ and $\zeta_t = \frac{\partial q^\psi}{\partial \psi}(t, X_t^{\boldsymbol{\pi}^\psi}, a_t^{\boldsymbol{\pi}^\psi})$, we find that the modified SARSA in Tallec et al. (2019) and Algorithm 3 differ only by a term whose mean is 0 due to the constraint (29) on the q-function. This term is solely driven by the action randomization at time $t+\Delta t$. Hence, the algorithm in Tallec et al. (2019) is noisier and may be slower in convergence speed compared with the q-learning one. In Section 7, we will numerically compare the results of the above Q-learning SARSA and q-learning.

5 q-Learning Algorithms When Normalizing Constant Is Unavailable

Theorem 7 assumes that the normalizing constant in the Gibbs measure is available so that one can enforce the constraint (21). So does Theorem 9 because the second constraint in (27) is exactly to verify the normalizing constant to be 1. But it is well known that in most high-dimensional cases computing this constant is a daunting, and often impossible task.

5.1 A stronger policy improvement theorem

To overcome the difficulty arising from an unavailable normalizing constant in soft Q-learning, Haarnoja et al. (2018b) propose a general method of using a family of stochastic policies whose densities can be easily computed to approximate $\boldsymbol{\pi}'$. In our current setting, denote by $\{\boldsymbol{\pi}^\phi(\cdot|t, x)\}_{\phi \in \Phi}$ the family of density functions of some tractable distributions, e.g., the multivariate normal distribution $\boldsymbol{\pi}^\phi(\cdot|t, x) = \mathcal{N}\left(\mu^\phi(t, x), \Sigma^\phi(t, x)\right)$, where $\mu^\phi(t, x) \in \mathbb{R}^m$ and $\Sigma^\phi(t, x) \in \mathbb{S}_{++}^m$. The learning procedure then proceeds as follows: starting from a policy $\boldsymbol{\pi}^\phi$ within this family, project the desired policy $\frac{\exp\{\frac{1}{\gamma}q(t,x,\cdot;\boldsymbol{\pi}^\phi)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}q(t,x,a;\boldsymbol{\pi}^\phi)\}\mathrm{d}a}$ to

the set $\{\boldsymbol{\pi}^{\phi}(a|t,x)\}_{\phi\in\Phi}$ by minimizing

$$\min_{\phi'\in\Phi} D_{KL}\left(\boldsymbol{\pi}^{\phi'}(\cdot|t,x)\Big\|\frac{\exp\{\frac{1}{\gamma}q(t,x,\cdot;\boldsymbol{\pi}^{\phi})\}}{\int_{\mathcal{A}}\exp\{\frac{1}{\gamma}q(t,x,a;\boldsymbol{\pi}^{\phi})\}\mathrm{d}a}\right) \equiv \min_{\phi'\in\Phi} D_{KL}\left(\boldsymbol{\pi}^{\phi'}(\cdot|t,x)\Big\|\exp\{\frac{1}{\gamma}q(t,x,\cdot;\boldsymbol{\pi}^{\phi})\}\right),$$

where $D_{KL}\left(f\big\|g\right) := \int_{\mathcal{A}} \log\frac{f(a)}{g(a)}f(a)\mathrm{d}a$ is the Kullback–Leibler (KL) divergence of two positive functions f, g with the same support on $\mathcal{A}$, where $f \in \mathcal{P}(\mathcal{A})$ is a probability density function on $\mathcal{A}$.

This procedure may not eventually lead to the desired target policy, but the newly defined policy still provably improves the previous policy, as indicated in the following stronger version of the policy improvement theorem that generalizes Theorem 2.

Theorem 10 *Given $(t,x) \in [0,T] \times \mathbb{R}^d$, if two policies $\boldsymbol{\pi} \in \Pi$ and $\boldsymbol{\pi}' \in \Pi$ satisfy*

$$D_{KL}\left(\boldsymbol{\pi}'(\cdot|t,x)\Big\|\exp\{\frac{1}{\gamma}H\big(t,x,\cdot,\frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}),\frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\}\right)$$

$$\leq D_{KL}\left(\boldsymbol{\pi}(\cdot|t,x)\Big\|\exp\{\frac{1}{\gamma}H\big(t,x,\cdot,\frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}),\frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\}\right),$$

then $J(t,x;\boldsymbol{\pi}') \geq J(t,x;\boldsymbol{\pi})$.

Theorem 10 in itself is a general result comparing two given policies not necessarily within any tractable family of densities. However, an implication of the result is that, given a current tractable policy $\boldsymbol{\pi}^{\phi}$, as long as we update it to a new tractable policy $\boldsymbol{\pi}^{\phi'}$ with

$$D_{KL}\left(\boldsymbol{\pi}^{\phi'}(\cdot|t,x)\Big\|\exp\{\frac{1}{\gamma}q(t,x,\cdot;\boldsymbol{\pi}^{\phi})\}\right) \leq D_{KL}\left(\boldsymbol{\pi}^{\phi}(\cdot|t,x)\Big\|\exp\{\frac{1}{\gamma}q(t,x,\cdot;\boldsymbol{\pi}^{\phi})\}\right),$$

then $\boldsymbol{\pi}^{\phi'}$ improves upon $\boldsymbol{\pi}^{\phi}$. Thus, to update $\boldsymbol{\pi}^{\phi}$ it suffices to solve the optimization problem

$$\min_{\phi'\in\Phi} D_{KL}\left(\boldsymbol{\pi}^{\phi'}(\cdot|t,x)\Big\|\exp\{\frac{1}{\gamma}q(t,x,\cdot;\boldsymbol{\pi}^{\phi})\}\right)$$

$$= \min_{\phi'\in\Phi}\int_{\mathcal{A}}\left[\log\boldsymbol{\pi}^{\phi'}(a|t,x) - \frac{1}{\gamma}q(t,x,a;\boldsymbol{\pi}^{\phi})\right]\boldsymbol{\pi}^{\phi'}(a|t,x)\mathrm{d}a.$$

The gradient of the above objective function in ϕ' is

$$\frac{\partial}{\partial\phi'}\int_{\mathcal{A}}\left[\log\boldsymbol{\pi}^{\phi'}(a|t,x) - \frac{1}{\gamma}q(t,x,a;\boldsymbol{\pi}^{\phi})\right]\boldsymbol{\pi}^{\phi'}(a|t,x)\mathrm{d}a$$

$$= \int_{\mathcal{A}}\left[\log\boldsymbol{\pi}^{\phi'}(a|t,x) - \frac{1}{\gamma}q(t,x,a;\boldsymbol{\pi}^{\phi})\right]\frac{\partial\boldsymbol{\pi}^{\phi'}(a|t,x)}{\partial\phi'}\mathrm{d}a + \int_{\mathcal{A}}\left[\frac{\partial}{\partial\phi'}\log\boldsymbol{\pi}^{\phi'}(a|t,x)\right]\boldsymbol{\pi}^{\phi'}(a|t,x)\mathrm{d}a$$

$$= \int_{\mathcal{A}}\left[\log\boldsymbol{\pi}^{\phi'}(a|t,x) - \frac{1}{\gamma}q(t,x,a;\boldsymbol{\pi}^{\phi})\right]\left[\frac{\partial}{\partial\phi'}\log\boldsymbol{\pi}^{\phi'}(a|t,x)\right]\boldsymbol{\pi}^{\phi'}(a|t,x)\mathrm{d}a,$$

where we have noted

$$\int_{\mathcal{A}}\left[\frac{\partial}{\partial\phi'}\log\boldsymbol{\pi}^{\phi'}(a|t,x)\right]\boldsymbol{\pi}^{\phi'}(a|t,x)\mathrm{d}a = \int_{\mathcal{A}}\frac{\partial}{\partial\phi'}\boldsymbol{\pi}^{\phi'}(a|t,x)\mathrm{d}a = \frac{\partial}{\partial\phi'}\int_{\mathcal{A}}\boldsymbol{\pi}^{\phi'}(a|t,x)\mathrm{d}a = 0.$$

Therefore, we may update ϕ each step incrementally by

$$\phi \leftarrow \phi - \gamma\alpha_\phi \mathrm{d}t \left[\log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|t, X_t) - \frac{1}{\gamma}q(t, X_t, a_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi) \right] \frac{\partial}{\partial\phi} \log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|t, X_t), \qquad (34)$$

where we choose the step size to be specifically $\gamma\alpha_\phi \mathrm{d}t$, for the reason that will become evident momentarily.

The above updating rule seems to indicate that we would need to learn the q-function $q(t, x, a; \boldsymbol{\pi}^\phi)$ associated with the policy $\boldsymbol{\pi}^\phi$. This is doable in view of Theorem 7, noting that $\{\boldsymbol{\pi}^\phi(\cdot|t, x)\}_{\phi\in\Phi}$ is a tractable family of distributions so that the constraint (21) or (29) can be computed and enforced. However, we do not need to do so given that (34) only requires the q-value along the trajectory $\{(t, X_t, a_t^{\boldsymbol{\pi}^\phi}); 0 \le t \le T\}$, instead of the full functional form $q(\cdot, \cdot, \cdot; \boldsymbol{\pi}^\phi)$. The former is easier to computed from the temporal difference learning, as will be evident in the next subsection.

5.2 Connections with policy gradient

Applying Itô's lemma to $J(\cdot, \cdot; \boldsymbol{\pi})$ and recalling Definition 4, we have the following important relation between the q-function and the temporal difference of the value function:

$$q(t, X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}; \boldsymbol{\pi})\mathrm{d}t = \mathrm{d}J(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + r(t, X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}})\mathrm{d}t - \beta J(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi})\mathrm{d}t + \{\cdots\}\mathrm{d}W_t. \qquad (35)$$

Plugging the above to (34) and ignoring the martingale difference term $\{\cdots\}\mathrm{d}W_t$ whose mean is 0, the gradient-based updating rule for ϕ becomes

$$\phi \leftarrow \phi + \alpha_\phi \left[-\gamma \log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|t, X_t^{\boldsymbol{\pi}^\phi})\mathrm{d}t + \mathrm{d}J(t, X_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi) + r(t, X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi})\mathrm{d}t - \beta J(t, X_t^{\boldsymbol{\pi}^\phi}; \boldsymbol{\pi}^\phi)\mathrm{d}t \right]$$
$$\times \frac{\partial}{\partial\phi} \log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|t, X_t^{\boldsymbol{\pi}^\phi}). \qquad (36)$$

In this updating rule, we only need to approximate $J(\cdot, \cdot; \boldsymbol{\pi}^\phi)$, which is a problem of PE. Specifically, denote by $J^\theta(\cdot, \cdot)$ the function approximator of $J(\cdot, \cdot; \boldsymbol{\pi}^\phi)$, where $\theta \in \Theta$, and we may apply any PE methods developed in Jia and Zhou (2022a) to learn J^θ. Given this value function approximation, the updating rule (36) further specializes to

$$\phi \leftarrow \phi + \alpha_\phi \left[-\gamma \log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|t, X_t^{\boldsymbol{\pi}^\phi})\mathrm{d}t + \mathrm{d}J^\theta(t, X_t^{\boldsymbol{\pi}^\phi}) + r(t, X_t^{\boldsymbol{\pi}^\phi}, a_t^{\boldsymbol{\pi}^\phi})\mathrm{d}t - \beta J^\theta(t, X_t^{\boldsymbol{\pi}^\phi})\mathrm{d}t \right]$$
$$\times \frac{\partial}{\partial\phi} \log \boldsymbol{\pi}^\phi(a_t^{\boldsymbol{\pi}^\phi}|t, X_t^{\boldsymbol{\pi}^\phi}). \qquad (37)$$

Expression (37) coincides with the updating rule in the *policy gradient* method established in Jia and Zhou (2022b) when the regularizer therein is taken to be the entropy. Moreover, since we learn the value function and the policy simultaneously, the updating rule leads to actor–critic type of algorithms. Finally, with this method, while learning the value function may involve martingale conditions as in Jia and Zhou (2022a), learning the policy does not.

Schulman et al. (2017) note the equivalence between soft Q-learning and PG in discrete time. Here we present the continuous-time counterpart of the equivalence, nevertheless with a theoretical justification, which recovers the PG based algorithms in Jia and Zhou (2022b) when combined with suitable PE methods.

6 Extension to Ergodic Tasks

We now extend the previous study to ergodic tasks, in which the objective is to maximize the long-run average in the infinite time horizon $[0, \infty)$, the functions b, σ and r do not depend on time t explicitly, and $h = 0$. The set of admissible (stationary) policies can be similarly defined in a straighforward manner. The regularized ergodic objective function is the long-run average:

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

where $\gamma \geq 0$ is the temperature parameter.

We first present the ergodic version of the Feynman–Kac formula and the corresponding policy improvement theorem.

Lemma 11 *Let $\boldsymbol{\pi} = \boldsymbol{\pi}(\cdot|\cdot)$ be a given admissible (stationary) policy.[14] Suppose there is a function $J(\cdot; \boldsymbol{\pi}) \in C^2(\mathbb{R}^d)$ with polynomial growth and a scalar $V(\boldsymbol{\pi}) \in \mathbb{R}$ satisfying*

$$
\int_{\mathcal{A}} \left[H\left(x, a, \frac{\partial J}{\partial x}(x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(x; \boldsymbol{\pi})\right) - \gamma \log \boldsymbol{\pi}(a|x) \right] \boldsymbol{\pi}(a|x) \mathrm{d}a - V(\boldsymbol{\pi}) = 0, \quad x \in \mathbb{R}^d. \quad (38)
$$

Then for any $t \geq 0$,

$$
\begin{aligned}
V(\boldsymbol{\pi}) &= \liminf_{T \to \infty} \tfrac{1}{T} \mathbb{E}^{\mathbb{P}^W} \left(\int_t^T \int_{\mathcal{A}} [r(\tilde{X}_s^{\boldsymbol{\pi}}, a) - \gamma \log \boldsymbol{\pi}(a|\tilde{X}_s^{\boldsymbol{\pi}})] \boldsymbol{\pi}(a|\tilde{X}_s^{\boldsymbol{\pi}}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x \right) \\
&= \liminf_{T \to \infty} \tfrac{1}{T} \mathbb{E}^{\mathbb{P}} \left(\int_t^T [r(X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_s^{\boldsymbol{\pi}}|X_s^{\boldsymbol{\pi}})] \mathrm{d}s \Big| X_t^{\boldsymbol{\pi}} = x \right).
\end{aligned}
$$
$$(39)$$

Moreover, if there are two admissible policies $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ such that for all $x \in \mathbb{R}^d$,

$$
D_{KL} \left(\boldsymbol{\pi}'(\cdot|x) \Big\| \exp\{\frac{1}{\gamma} H\left(x, \cdot, \frac{\partial J}{\partial x}(x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(x; \boldsymbol{\pi}))\} \right)
$$
$$
\leq D_{KL} \left(\boldsymbol{\pi}(\cdot|x) \Big\| \exp\{\frac{1}{\gamma} H\left(x, \cdot, \frac{\partial J}{\partial x}(x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(x; \boldsymbol{\pi}))\} \right),
$$

then $V(\boldsymbol{\pi}') \geq V(\boldsymbol{\pi})$.

We emphasize that the solution to (38) 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 ergodicity, and hence is a constant as (39) implies. The function J, on the other hand, represents the first-order approximation of the long-run average and is not unique. Indeed, for any constant c, $(J + c, V)$ is also a solution to (38). We refer to V as the *value* of the underlying problem and still refer to J as the

---

14. For such ergodic tasks, typically an admissible policies also require the process $X^{\boldsymbol{\pi}}$ is ergodic (cf. Meyn and Tweedie 1993).

value function. 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 learning tasks.

As with the episodic case, for any admissible policy $\boldsymbol{\pi}$, we can define the Δt-parameterized Q-function as

$$
\begin{aligned}
&Q_{\Delta t}(x, a; \boldsymbol{\pi}) \\
=&\mathbb{E}^{\mathbb{P}^W}\bigg[\int_t^{t+\Delta t}[r(X_s^a, a) - V(\boldsymbol{\pi})]\mathrm{d}s + \lim_{T\to\infty}\mathbb{E}^{\mathbb{P}}\big[\int_{t+\Delta t}^T [r(X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \gamma\log\boldsymbol{\pi}(a_s^{\boldsymbol{\pi}}|X_s^{\boldsymbol{\pi}}) \\
&- V(\boldsymbol{\pi})]\mathrm{d}s|X_{t+\Delta t}^a\big]\Big|X_t^{\boldsymbol{\pi}} = x\bigg] \\
=&\mathbb{E}^{\mathbb{P}^W}\bigg[\int_t^{t+\Delta t}[r(X_s^a, a) - V(\boldsymbol{\pi})]\mathrm{d}s + J(X_{t+\Delta t}^a; \boldsymbol{\pi})\Big|X_t^{\boldsymbol{\pi}} = x\bigg] \\
&- \mathbb{E}^{\mathbb{P}^W}\bigg[\lim_{T\to\infty}\mathbb{E}^{\mathbb{P}}\big[J(X_T^{\boldsymbol{\pi}}; \boldsymbol{\pi})|X_{t+\Delta t}^a\big]\Big|X_t^{\boldsymbol{\pi}} = x\bigg] \\
=&\mathbb{E}^{\mathbb{P}^W}\bigg[\int_t^{t+\Delta t}[r(X_s^a, a) - V(\boldsymbol{\pi})]\mathrm{d}s + J(X_{t+\Delta t}^a; \boldsymbol{\pi}) - J(x; \boldsymbol{\pi})\Big|X_t^{\boldsymbol{\pi}} = x\bigg] + J(x; \boldsymbol{\pi}) \\
&- \mathbb{E}^{\mathbb{P}^W}\bigg[\lim_{T\to\infty}\mathbb{E}^{\mathbb{P}}\big[J(X_T^{\boldsymbol{\pi}}; \boldsymbol{\pi})|X_{t+\Delta t}^a\big]\Big|X_t^{\boldsymbol{\pi}} = x\bigg] \\
=&\mathbb{E}^{\mathbb{P}^W}\bigg[\int_t^{t+\Delta t}\big[H\big(X_s^a, a, \frac{\partial J}{\partial x}(X_s^a; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(X_s^a; \boldsymbol{\pi})\big) - V(\boldsymbol{\pi})]\mathrm{d}s\Big|X_t^{\boldsymbol{\pi}} = x\bigg] + J(x; \boldsymbol{\pi}) \\
&- \mathbb{E}^{\mathbb{P}^W}\bigg[\lim_{T\to\infty}\mathbb{E}^{\mathbb{P}}\big[J(X_T^{\boldsymbol{\pi}}; \boldsymbol{\pi})|X_{t+\Delta t}^a\big]\Big|X_t^{\boldsymbol{\pi}} = x\bigg] \\
=&J(x; \boldsymbol{\pi}) + \bigg[H\left(x, a, \frac{\partial J}{\partial x}(x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(x; \boldsymbol{\pi})\right) - V(\boldsymbol{\pi})\bigg]\Delta t - \bar{J} + O\big((\Delta t)^2\big),
\end{aligned}
$$

where we assumed that $\lim_{T\to\infty}\mathbb{E}^{\mathbb{P}}\big[J(X_T^{\boldsymbol{\pi}}; \boldsymbol{\pi})|X_t^{\boldsymbol{\pi}} = x\big] = \bar{J}$ is a constant for any $(t, x) \in [0, +\infty) \times \mathbb{R}^d$.

The corresponding q-function is then defined as

$$
q(x, a; \boldsymbol{\pi}) = H\left(x, a, \frac{\partial J}{\partial x}(x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(x; \boldsymbol{\pi})\right) - V(\boldsymbol{\pi}).
$$

As a counterpart to Theorem 7, we have the following theorem to characterize the value function, the value, and the q-function for both on-policy and off-policy ergodic learning problems. A counterpart to Theorem 9 can be similarly established, and we leave details to the reader.

Theorem 12 *Let an admissible policy $\boldsymbol{\pi}$, a number $\hat{V}$, a function $\hat{J} \in C^2(\mathbb{R}^d)$ with polynomial growth, and a continuous function $\hat{q} : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ be given satisfying*

$$
\lim_{T\to\infty}\frac{1}{T}\mathbb{E}[\hat{J}(X_T^{\boldsymbol{\pi}})] = 0, \quad \int_{\mathcal{A}}\big[\hat{q}(x, a) - \gamma\log\boldsymbol{\pi}(a|x)\big]\boldsymbol{\pi}(a|x)\mathrm{d}a = 0, \quad \forall x \in \mathbb{R}^d. \tag{40}
$$

Then

29

(i) $\hat{V}$, $\hat{J}$ and $\hat{q}$ are respectively the value, the value function and the q-function associated with $\boldsymbol{\pi}$ if and only if for all $x \in \mathbb{R}^d$, the following process

$$\hat{J}(X_t^{\boldsymbol{\pi}}) + \int_0^t [r(X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \hat{q}(X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \hat{V}]\mathrm{d}s \qquad (41)$$

is an $(\{\mathcal{F}_t\}_{t \geq 0}, \mathbb{P})$-martingale, where $\{X_t^{\boldsymbol{\pi}}, 0 \leq t < \infty\}$ is the solution to (6) with $X_0^{\boldsymbol{\pi}} = x$.

(ii) If $\hat{V}$, $\hat{J}$ and $\hat{q}$ are respectively the value, value function and the q-function associated with $\boldsymbol{\pi}$, then for any admissible $\boldsymbol{\pi}'$ and any $x \in \mathbb{R}^d$, the following process

$$\hat{J}(X_t^{\boldsymbol{\pi}'}) + \int_0^t [r(X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'}) - \hat{q}(X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'}) - \hat{V}]\mathrm{d}u \qquad (42)$$

is an $(\{\mathcal{F}_t\}_{t \geq 0}, \mathbb{P})$-martingale, where $\{X_t^{\boldsymbol{\pi}'}, 0 \leq t < \infty$ is the solution to (6) under $\boldsymbol{\pi}'$ with initial condition $X_0^{\boldsymbol{\pi}'} = x$.

(iii) If there exists an admissible $\boldsymbol{\pi}'$ such that for all $x \in \mathbb{R}^d$, (42) is an $(\{\mathcal{F}_t\}_{t \geq 0}, \mathbb{P})$-martingale where $X_0^{\boldsymbol{\pi}'} = x$, then $\hat{V}$, $\hat{J}$ and $\hat{q}$ are respectively the value, value function and the q-function associated with $\boldsymbol{\pi}$.

Moreover, in any of the three cases above, if it holds further that $\boldsymbol{\pi}(a|x) = \frac{\exp\{\frac{1}{\gamma}\hat{q}(x,a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}\hat{q}(x,a)\}\mathrm{d}a}$, then $\boldsymbol{\pi}$ is the optimal policy and $\hat{V}$ is the optimal value.

Based on Theorem 12, we can design corresponding q-learning algorithms for ergodic problems, in which we learn V, J^θ and q^ψ at the same time. These algorithms have natural connections with SARSA as well as the PG-based algorithms developed in Jia and Zhou (2022b). The discussions are similar to those in Subsections 4.2 and 5.2 and hence are omitted here. An online algorithm for ergodic tasks is presented as an example; see Algorithm 4.

7 Applications

7.1 Mean–variance portfolio selection

We first review the formulation of the exploratory mean–variance portfolio selection problem, originally proposed by Wang and Zhou (2020) and later revisited by Jia and Zhou (2022b).[15] 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 $\{S_t, 0 \leq t \leq T\}$ is governed by a geometric Brownian motion with drift μ and volatility $\sigma > 0$, defined on a filtered probability space $(\Omega, \mathcal{F}, \mathbb{P}^W; \{\mathcal{F}_t^W\}_{0 \leq t \leq T})$. 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 $a = \{a_t, 0 \leq t \leq T\}$, where a_t

---

15. There is a vast literature on classical model-based (non-exploratory) continuous-time mean–variance models formulated as stochastic control problems; see e.g. Zhou and Li (2000); Lim and Zhou (2002); Zhou and Yin (2003) and the references therein.

Algorithm 4 q-Learning Algorithm for Ergodic Tasks

Inputs: initial state x_0, time step Δt, initial learning rates $\alpha_\theta, \alpha_\psi, \alpha_V$ and learning rate schedule function $l(\cdot)$ (a function of time), functional forms of the parameterized value function $J^\theta(\cdot)$ and q-function $q^\psi(\cdot, \cdot)$ satisfying (40), functional forms of test functions $\boldsymbol{\xi}(t, x_{\cdot \wedge t}, a_{\cdot \wedge t})$, $\boldsymbol{\zeta}(t, x_{\cdot \wedge t}, a_{\cdot \wedge t})$, and temperature parameter γ.

Required program (on-policy): an environment simulator $(x', r) = Environment_{\Delta t}(x, a)$ that takes initial state x and action a as inputs and generates a new state x' at Δt and an instantaneous reward r as outputs. Policy $\boldsymbol{\pi}^\psi(a|x) = \frac{\exp\{\frac{1}{\gamma}q^\psi(x,a)\}}{\int_\mathcal{A} \exp\{\frac{1}{\gamma}q^\psi(x,a)\}\mathrm{d}a}$.

Required program (off-policy): observations $\{a, r, x'\} = Observation(x; \Delta t)$ including the observed actions, rewards, and state when the current state is x under the given behavior policy at the sampling time grids with step size Δt.

Learning procedure:

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

  loop

    {**On-policy case**

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

    Apply a to environment simulator $(x', r) = Environment_{\Delta t}(x, a)$, and observe new state x' and reward r as outputs. Store $x_{t_{k+1}} \leftarrow x'$.

    }

    {**Off-policy case**

    Obtain one observation $a_{t_k}, r_{t_k}, x_{t_{k+1}} = Observation(x_{t_k}; \Delta t)$.

    }

    Compute test functions $\xi_{t_k} = \boldsymbol{\xi}(t_k, x_{t_0}, \cdots, x_{t_k}, a_{t_0}, \cdots, a_{t_k})$, $\zeta_{t_k} = \boldsymbol{\zeta}(t_k, x_{t_0}, \cdots, x_{t_k}, a_{t_0}, \cdots, a_{t_k})$.

    Compute

$$\delta = J^\theta(x') - J^\theta(x) + r\Delta t - q^\psi(x, a)\Delta t - V\Delta t,$$
$$\Delta\theta = \xi_{t_k}\delta,$$
$$\Delta V = \delta,$$
$$\Delta\psi = \zeta_{t_k}\delta.$$

  Update θ, V and ψ by

$$\theta \leftarrow \theta + l(k\Delta t)\alpha_\theta \Delta\theta,$$
$$V \leftarrow V + l(k\Delta t)\alpha_V \Delta V,$$
$$\psi \leftarrow \psi + l(k\Delta t)\alpha_\psi \Delta\psi.$$

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

  end loop

---

is the discounted dollar value invested in the risky asset at time t. Then her discounted wealth process follows

$$\mathrm{d}X_t = a_t[(\mu - r)\mathrm{d}t + \sigma\mathrm{d}W_t] = a_t\frac{\mathrm{d}(e^{-rt}S_t)}{e^{-rt}S_t}, \quad X_0 = x_0.$$

The agent aims to minimize the variance of the discounted value of the portfolio at time T while maintaining the expected return to be a certain level; that is,

$$\min_a \mathrm{Var}(X_T^a), \text{ subject to } \mathbb{E}[X_T^a] = z, \tag{43}$$

where z is the target expected return, and the variance and expectation throughout this subsection are with respect to the probability measure $\mathbb{P}^W$.

The exploratory formulation of this problem with entropy regularizer is equivalent to

$$J(t, x; w) = -\max_{\boldsymbol{\pi}} \mathbb{E}\left[-(\tilde{X}_T^{\boldsymbol{\pi}} - w)^2 - \gamma \int_t^T \log \boldsymbol{\pi}(a_s^{\boldsymbol{\pi}}|s, \tilde{X}^{\boldsymbol{\pi}})\mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}} = x \right] + (w - z)^2, \tag{44}$$

subject to

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

where w is the Lagrange multiplier introduced to relax the expected return constraint; see Wang and Zhou (2020) for a derivation of this formulation. Note here we artificially add the minus sign to transform the variance minimization to a maximization problem to be consistent with our general formulation.

We now follow Wang and Zhou (2020) and Jia and Zhou (2022b) to parameterize the value function by

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

Moreover, we parameterize the q-function by

$$q^\psi(t, x, a; w) = -\frac{e^{-\psi_1 - \psi_3(T-t)}}{2}\big(a + \psi_2(x - w)\big)^2 - \frac{\gamma}{2}[\log 2\pi\gamma + \psi_1 + \psi_3(T - t)],$$

which is derived to satisfy the constraint (29), as explained in Example 1. The policy associated with this parametric q-function is $\boldsymbol{\pi}^\psi(\cdot|x; w) = \mathcal{N}(-\psi_2(x - w), \gamma e^{\psi_1 + \psi_3(T-t)})$. In addition to $\theta = (\theta_1, \theta_2, \theta_3)^\top$ and $\psi = (\psi_1, \psi_2, \psi_3)^\top$, we also learn the Lagrange multiplier w by stochastic approximation in the same way as in Wang and Zhou (2020) and Jia and Zhou (2022b). The full algorithm is summarized in Algorithm 5.[16]

We then compare by simulations the performances of three learning algorithms: the q-learning based Algorithm 5 in this paper, the PG-based Algorithm 4 in Jia and Zhou (2022b), and the Δt-parameterized Q-learning algorithm presented in Appendix B. We conduct simulations with the following configurations: $\mu \in \{0, \pm 0.1, \pm 0.3, \pm 0.5\}$, $\sigma \in \{0.1, 0.2, 0.3, 0.4\}$, $T = 1$, $x_0 = 1$, $z = 1.4$. Other tuning parameters in all the algorithms are chosen as $\gamma = 0.1$, $m = 10$, $\alpha_\theta = \alpha_\psi = 0.001$, $\alpha_w = 0.005$, and $l(j) = \frac{1}{j^{0.51}}$. To have more realistic scenarios, we generate 20 years of training data and compare the three algorithms with the same dataset for $N = 20,000$ episodes with a batch size 32. More precisely, 32 trajectories with length T are drawn from the training set to update the parameters to

---

16. Here we present an offline algorithm as example. Online algorithms can also be devised following the general study in the previous sections.

Algorithm 5 Offline–Episodic q-Learning 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 $\alpha_\theta, \alpha_\psi, \alpha_w$ 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, ψ, 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**

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

      Compute and store the test function $\xi_{t_k} = \frac{\partial J^\theta}{\partial \theta}(t_k, x_{t_k}; w)$, $\zeta_{t_k} = \frac{\partial q^\psi}{\partial \psi}(t_k, x_{t_k}, a_{t_k}; w)$.

      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

$$\Delta \theta = \sum_{i=0}^{K-1} \xi_{t_i} \left[J^\theta(t_{i+1}, x_{t_{i+1}}; w) - J^\theta(t_i, x_{t_i}; w) - q^\psi(t_i, x_{t_i}, a_{t_i}; w)\Delta t \right],$$

$$\Delta \psi = \sum_{i=0}^{K-1} \zeta_{t_i} \left[J^\theta(t_{i+1}, x_{t_{i+1}}; w) - J^\theta(t_i, x_{t_i}; w) - q^\psi(t_i, x_{t_i}, a_{t_i}; w)\Delta t \right].$$

  Update θ and ψ by

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

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

    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

---

be learned for N times. After training completes, for each market configuration we apply the learned policy out-of-sample repeatedly for 100 times and compute the average mean, variance and Sharpe ratio for each algorithm. We are particularly interested in the impact of time discretization; so we experiment with three different time discretization steps: $\Delta t \in \{\frac{1}{25}, \frac{1}{250}, \frac{1}{2500}\}$. Note that all the algorithms rely on time discretization at the implementation stage, where Δt is the sampling frequency required for execution or computation of numerical integral for PG and q-learning. For the Q-learning, Δt plays a dual role: it is both the parameter in its definition and the time discretization size in implementation.

The numerical results are reported in Tables 1 – 3, each table corresponding to a different value of Δt. We observe that for any market configuration, the Q-learning is almost always the worst performer in terms of all the metrics, and even divergent in certain high volatility and low return environment (when keeping the same learning rate as in the other cases). The other two algorithms have very close overall performances, although the q-learning tends to outperform in high volatile market environments. On the other hand, notably, the results change only slightly with different time discretizations, for *all* the three algorithms. This observation does not align with the claim in Tallec et al. (2019) that the classical Q-learning is very sensitive to time-discretization, at least for this specific application problem. In addition, we notice that q-learning and PG-based algorithm produce comparable results under most market scenarios, while the terminal variance produced by q-learning tends to be significantly higher than that by PG when $|\mu|$ is small or σ is large. In such cases, the ground truth solution requires higher leverage in risky assets in order to reach the high target expected return (40% annual return). However, it appears that overall q-learning strives to learn to keep up with the high target return and as a result ends up with high terminal variance in those cases. By contrast, the PG tends to maintain lower variance at the cost of significantly underachieving the target return.

7.2 Ergodic linear–quadratic control

Consider the ergodic LQ control problem where state responds to actions in a linear way

$$dX_t = (AX_t + Ba_t)dt + (CX_t + Da_t)dW_t, \ X_0 = x_0,$$

and the goal is to maximize the long term average quadratic payoff

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

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

The exploratory formulation of this problem with entropy regularizer is equivalent to

$$\liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\int_0^T r(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}})dt - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}}|X_t^{\boldsymbol{\pi}})dt \Big| X_0^{\boldsymbol{\pi}} = x_0\right].$$

This problem falls into the general formulation of an ergodic task studied in Section 6; so we directly implement Algorithm 4 in our simulation. In particular, our function approximators for J and q are quadratic functions:

$$J^\theta(x) = \theta_1 x^2 + \theta_2 x, \quad q^\psi(x,a) = -\frac{e^{-\psi_3}}{2}(a - \psi_1 x - \psi_2)^2 - \frac{\gamma}{2}(\log 2\pi\gamma + \psi_3),$$

34

Table 1: **Out-of-sample performance comparison in terms of mean, variance and Sharpe ratio when data are generated by geometric Brownian motion with $\Delta t = \frac{1}{25}$.** We compare three algorithms: "Policy Gradient" described in Algorithm 4 in Jia and Zhou (2022b), "Q-Learning" described in Appendix B, and "q-Learning" described in Algorithm 5. The first two columns specify market configurations (parameters of the geometric Brownian motion simulated). "Mean", "Variance", and "SR" refer respectively to the average mean, variance, and Sharpe ratio of the terminal wealth under the corresponding learned policy over 100 independent runs. We highlight the largest average Sharpe ratio in **bold**. "NA" refers to the case where the algorithm diverges. The discretization size is $\Delta t = \frac{1}{25}$.

μ	σ	Policy Gradient			Q-Learning			q-Learning		
		Mean	Variance	SR	Mean	Variance	SR	Mean	Variance	SR
-0.5	0.1	1.409	0.003	**7.134**	1.409	0.004	6.663	1.408	0.004	6.805
-0.3	0.1	1.421	0.012	**3.880**	1.409	0.012	3.707	1.412	0.012	3.762
-0.1	0.1	1.382	0.094	**1.285**	1.327	0.070	1.263	1.351	0.081	1.272
0.0	0.1	1.090	0.216	0.193	1.098	0.302	0.201	1.108	0.321	**0.202**
0.1	0.1	1.312	0.147	**0.835**	1.265	0.108	0.827	1.292	0.133	0.831
0.3	0.1	1.420	0.016	**3.312**	1.402	0.016	3.181	1.407	0.016	3.224
0.5	0.1	1.405	0.004	**6.422**	1.403	0.005	6.026	1.403	0.004	6.147
-0.5	0.2	1.417	0.014	**3.526**	1.416	0.016	3.308	1.416	0.016	3.377
-0.3	0.2	1.445	0.060	**1.915**	1.428	0.059	1.841	1.434	0.059	1.867
-0.1	0.2	1.335	0.341	0.608	1.402	0.801	0.628	1.405	0.589	**0.631**
0.0	0.2	1.053	0.539	0.070	1.168	7.995	0.100	1.136	3.422	**0.100**
0.1	0.2	1.248	0.437	0.380	1.349	2.046	0.411	1.341	1.005	**0.412**
0.3	0.2	1.445	0.083	**1.634**	1.422	0.078	1.580	1.431	0.081	1.600
0.5	0.2	1.413	0.018	**3.173**	1.412	0.020	2.992	1.411	0.019	3.050
-0.5	0.3	1.433	0.039	**2.305**	1.432	0.043	2.180	1.432	0.041	2.223
-0.3	0.3	1.458	0.162	**1.247**	1.468	0.196	1.214	1.484	0.217	1.229
-0.1	0.3	1.226	0.569	0.335	1.624	8.959	0.415	1.556	5.558	**0.415**
0.0	0.3	1.030	0.777	0.036	NA	NA	NA	1.187	23.269	**0.066**
0.1	0.3	1.151	0.670	0.202	1.539	13.741	0.271	1.482	13.321	**0.271**
0.3	0.3	1.448	0.231	1.046	1.475	0.322	1.042	1.485	0.313	**1.053**
0.5	0.3	1.431	0.048	**2.073**	1.428	0.052	1.972	1.429	0.050	2.008
-0.5	0.4	1.457	0.093	**1.679**	1.460	0.102	1.608	1.462	0.102	1.638
-0.3	0.4	1.417	0.346	0.877	1.670	2.363	0.898	1.602	1.176	**0.905**
-0.1	0.4	1.147	0.805	0.205	NA	NA	NA	1.711	20.852	**0.306**
0.0	0.4	1.017	0.893	0.016	NA	NA	NA	1.211	76.040	**0.048**
0.1	0.4	1.084	0.825	0.109	NA	NA	NA	1.539	25.371	**0.196**
0.3	0.4	1.408	0.379	0.747	NA	NA	NA	1.609	1.992	**0.776**
0.5	0.4	1.446	0.102	**1.510**	1.456	0.125	1.455	1.461	0.128	1.479

Table 2: **Out-of-sample performance comparison in terms of mean, variance and Sharpe ratio when data are generated by geometric Brownian motion with** $\Delta t = \frac{1}{250}$**.** We compare three algorithms: "Policy Gradient" described in Algorithm 4 in Jia and Zhou (2022b), "Q-Learning" described in Appendix B, and "q-Learning" described in Algorithm 5. The first two columns specify market configurations (parameters of the geometric Brownian motion simulated). "Mean", "Variance", and "SR" refer respectively to the average mean, variance, and Sharpe ratio of the terminal wealth under the corresponding learned policy over 100 independent runs. We highlight the largest average Sharpe ratio in **bold**. "NA" refers to the case where the algorithm diverges. The discretization size is $\Delta t = \frac{1}{250}$.

μ	σ	Policy Gradient			Q-Learning			q-Learning		
		Mean	Variance	SR	Mean	Variance	SR	Mean	Variance	SR
-0.5	0.1	1.409	0.003	**7.130**	1.407	0.004	6.671	1.407	0.004	6.805
-0.3	0.1	1.419	0.012	**3.878**	1.407	0.012	3.710	1.410	0.012	3.762
-0.1	0.1	1.380	0.092	**1.285**	1.325	0.069	1.264	1.347	0.079	1.272
0	0.1	1.092	0.218	**0.201**	1.096	0.253	0.201	1.106	0.306	0.202
0.1	0.1	1.314	0.148	**0.835**	1.267	0.109	0.827	1.293	0.133	0.831
0.3	0.1	1.424	0.017	**3.311**	1.406	0.017	3.183	1.411	0.017	3.224
0.5	0.1	1.411	0.004	**6.421**	1.409	0.005	6.032	1.409	0.004	6.147
-0.5	0.2	1.415	0.014	**3.524**	1.414	0.016	3.312	1.413	0.015	3.377
-0.3	0.2	1.439	0.057	**1.915**	1.423	0.056	1.843	1.428	0.056	1.867
-0.1	0.2	1.343	0.329	0.620	1.369	0.460	0.628	1.390	0.493	**0.631**
0	0.2	1.054	0.546	0.070	1.173	9.036	0.100	1.128	2.519	**0.100**
0.1	0.2	1.241	0.413	0.372	1.350	1.356	0.411	1.340	0.925	**0.412**
0.3	0.2	1.450	0.084	**1.634**	1.426	0.079	1.581	1.435	0.081	1.600
0.5	0.2	1.420	0.018	**3.173**	1.418	0.020	2.995	1.418	0.019	3.050
-0.5	0.3	1.428	0.037	**2.304**	1.426	0.041	2.183	1.426	0.039	2.223
-0.3	0.3	1.467	0.171	**1.248**	1.454	0.169	1.215	1.466	0.178	1.229
-0.1	0.3	1.230	0.552	0.326	1.661	10.321	0.415	1.495	2.852	**0.415**
0	0.3	1.031	0.845	0.036	1.236	38.868	0.066	1.184	21.710	**0.066**
0.1	0.3	1.142	0.716	0.181	1.541	16.335	0.266	1.456	6.433	**0.271**
0.3	0.3	1.466	0.230	**1.064**	1.470	0.260	1.043	1.486	0.281	1.053
0.5	0.3	1.438	0.049	**2.073**	1.434	0.053	1.974	1.435	0.051	2.008
-0.5	0.4	1.451	0.085	**1.679**	1.448	0.090	1.611	1.449	0.087	1.638
-0.3	0.4	1.424	0.281	0.890	1.535	0.638	0.898	1.544	0.612	**0.905**
-0.1	0.4	1.153	0.777	0.192	NA	NA	NA	1.747	22.299	**0.306**
0	0.4	1.019	1.000	0.021	NA	NA	NA	1.233	77.875	**0.048**
0.1	0.4	1.091	1.006	0.110	NA	NA	NA	1.696	48.274	**0.200**
0.3	0.4	1.400	0.384	0.704	1.685	2.737	0.771	1.595	1.189	**0.776**
0.5	0.4	1.463	0.114	**1.510**	1.463	0.122	1.457	1.466	0.121	1.479

Table 3: **Out-of-sample performance comparison in terms of mean, variance and Sharpe ratio when data are generated by geometric Brownian motion with** $\Delta t = \frac{1}{2500}$. We compare three algorithms: "Policy Gradient" described in Algorithm 4 in Jia and Zhou (2022b), "Q-Learning" described in Appendix B, and "q-Learning" described in Algorithm 5. The first two columns specify market configurations (parameters of the geometric Brownian motion simulated). "Mean", "Variance", and "SR" refer respectively to the average mean, variance, and Sharpe ratio of the terminal wealth under the corresponding learned policy over 100 independent runs. We highlight the largest average Sharpe ratio in **bold**. "NA" refers to the case where the algorithm diverges. The discretization size is $\Delta t = \frac{1}{2500}$.

| μ	σ	Policy Gradient			Q-Learning			q-Learning		
		Mean	Variance	SR	Mean	Variance	SR	Mean	Variance	SR
-0.5	0.1	1.408	0.003	**7.130**	1.406	0.004	6.677	1.406	0.004	6.805
-0.3	0.1	1.418	0.012	**3.878**	1.406	0.012	3.713	1.409	0.012	3.762
-0.1	0.1	1.376	0.090	**1.285**	1.324	0.069	1.264	1.345	0.077	1.272
0.0	0.1	1.092	0.218	0.201	1.096	0.259	0.201	1.105	0.300	**0.202**
0.1	0.1	1.316	0.149	**0.835**	1.269	0.111	0.827	1.294	0.133	0.831
0.3	0.1	1.425	0.017	**3.311**	1.407	0.017	3.185	1.412	0.017	3.224
0.5	0.1	1.412	0.004	**6.420**	1.410	0.005	6.037	1.410	0.004	6.147
-0.5	0.2	1.413	0.014	**3.524**	1.411	0.016	3.315	1.411	0.015	3.377
-0.3	0.2	1.435	0.056	**1.915**	1.419	0.055	1.844	1.424	0.056	1.867
-0.1	0.2	1.328	0.295	**0.632**	1.366	0.455	0.629	1.384	0.486	0.631
0.0	0.2	1.059	0.610	0.074	1.174	9.894	0.100	1.125	2.521	**0.100**
0.1	0.2	1.254	0.468	0.380	1.360	1.867	0.411	1.340	0.952	**0.412**
0.3	0.2	1.450	0.084	**1.634**	1.428	0.080	1.582	1.436	0.081	1.600
0.5	0.2	1.422	0.018	**3.173**	1.419	0.020	2.998	1.419	0.019	3.050
-0.5	0.3	1.425	0.036	**2.304**	1.422	0.040	2.185	1.422	0.039	2.223
-0.3	0.3	1.459	0.169	**1.248**	1.449	0.167	1.216	1.459	0.177	1.229
-0.1	0.3	1.225	0.597	0.327	1.690	11.223	0.415	1.487	2.943	**0.415**
0.0	0.3	1.039	0.857	0.044	1.206	36.278	0.066	1.183	23.798	**0.066**
0.1	0.3	1.169	0.745	0.202	1.457	14.390	0.266	1.462	8.590	**0.271**
0.3	0.3	1.455	0.215	**1.064**	1.488	0.356	1.044	1.486	0.296	1.053
0.5	0.3	1.438	0.049	**2.073**	1.435	0.053	1.975	1.436	0.051	2.008
-0.5	0.4	1.446	0.084	**1.679**	1.443	0.089	1.612	1.444	0.086	1.638
-0.3	0.4	1.395	0.231	0.873	1.529	0.619	0.899	1.537	0.622	**0.905**
-0.1	0.4	1.175	0.846	0.229	NA	NA	NA	1.754	24.745	**0.306**
0.0	0.4	1.026	0.967	0.025	NA	NA	NA	1.170	1.66E+09	**0.048**
0.1	0.4	1.110	0.973	0.129	NA	NA	NA	1.661	59.872	**0.200**
0.3	0.4	1.402	0.416	0.720	1.646	3.402	0.771	1.607	1.743	**0.776**
0.5	0.4	1.456	0.108	**1.510**	1.469	0.135	1.458	1.467	0.128	1.479
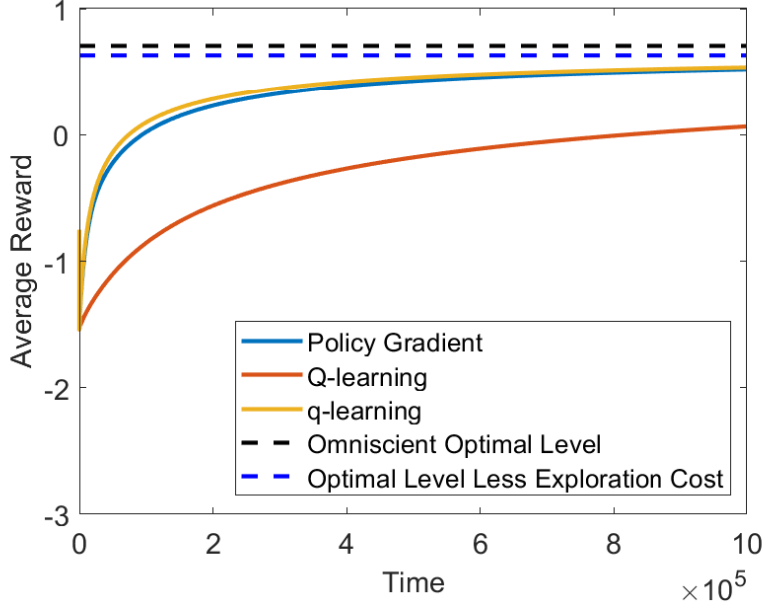
Figure 1: **Running average rewards of three RL algorithms.** A single state trajectory is generated with length $T = 10^6$ and discretized at $\Delta t = 0.1$ to which three online algorithms apply: "Policy Gradient" described in Algorithm 3 in Jia and Zhou (2022b), "Q-Learning" described in Appendix B, and "q-Learning" described in Algorithm 4. We repeat the experiments for 100 times for each method and plot the average reward over time with the shaded area indicating standard deviation. 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.

where the form of the q-function is derived to satisfy the constraint (29), as discussed in Example 1. The policy associated with this parameterized q-function is $\boldsymbol{\pi}^\psi(\cdot|x) = \mathcal{N}(\psi_1 x + \psi_2, \gamma e^{\psi_3})$. In addition, we have another parameter V that stands for the long-term average.

We then compare our online q-learning algorithm against two theoretical benchmarks and two other online algorithms, in terms of the running average reward during the learning process. The first benchmark is the omniscient optimal level, namely, the maximum long term average reward that can be achieved with perfect knowledge about the environment and reward (and hence exploration is unnecessary and only deterministic policies are considered). 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 agent who is however forced to explore under entropy regularization. The other two algorithms are the PG-based ergodic algorithm proposed in (Jia and Zhou, 2022b, Algorithm 3) and the Δt-based Q-learning algorithm presented in Appendix B.

In our simulation, to ensure 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 rates for all algorithms are initialized as $\alpha_\psi = 0.001$, and decay

(a) The path of learned ψ_1.   (b) The path of learned ψ_2.   (c) The path of learned e^{ψ_3}.
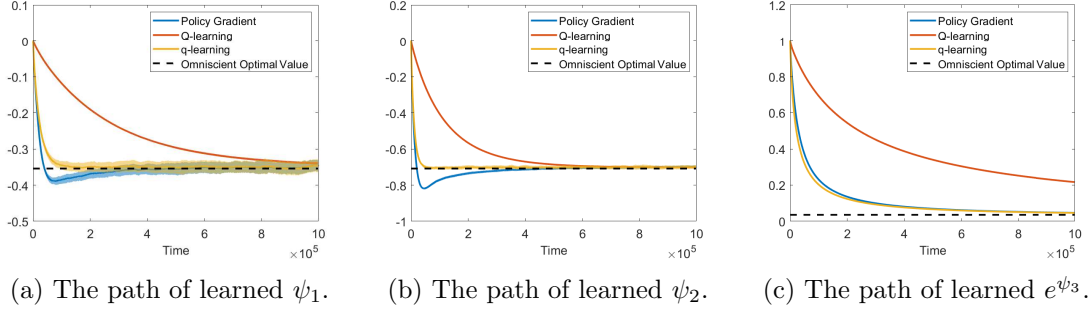
Figure 2: **Paths of learned parameters of three RL algorithms.** A single state trajectory is generated with length $T = 10^6$ and discretized at $\Delta t = 0.1$ to which three online algorithms apply: "Policy Gradient" described in Algorithm 3 in Jia and Zhou (2022b), "Q-Learning" described in Appendix B, and "q-Learning" described in Algorithm 4. All the policies are restricted to be in the parametric form of $\boldsymbol{\pi}^{\psi}(\cdot|x) = \mathcal{N}(\psi_1 x + \psi_2, e^{\psi_3})$. The omniscient optimal policy is $\psi_1^* \approx -0.354$, $\psi_2^* \approx -0.708$, $e^{\psi_3^*} \approx 0.035$, shown in the dashed line. We repeat the experiments for 100 times for each method and plot as the shaded area the standard deviation of the learned parameters. The width of each shaded area is twice the corresponding standard deviation.
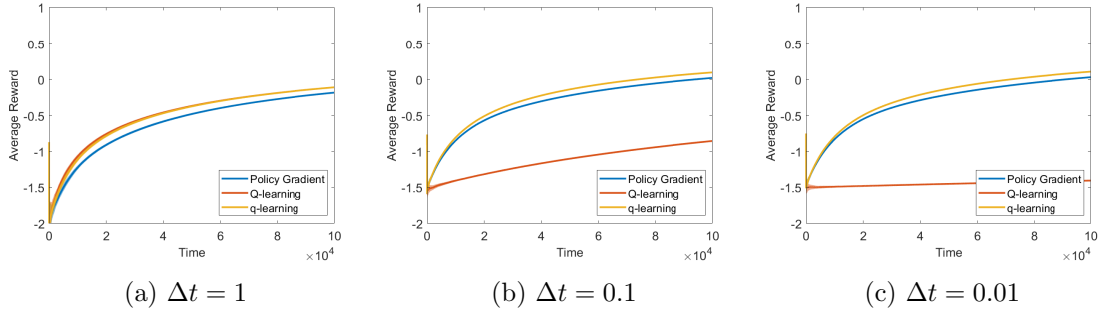

(a) $\Delta t = 1$ (b) $\Delta t = 0.1$ (c) $\Delta t = 0.01$

Figure 3: **Running average rewards of three RL algorithms with different time discretization sizes.** A single state trajectory is generated with length $T = 10^5$ and discretized at different step sizes: $\Delta t = 1$ in (a), $\Delta t = 0.1$ in (b), and $\Delta t = 0.01$ in (c). For each step size, we apply three online algorithms: "Policy Gradient" described in Algorithm 3 in Jia and Zhou (2022b), "Q-Learning" described in Appendix B, and "q-Learning" described in Algorithm 4. We repeat the experiments for 100 times for each method and plot the average reward over time with the shaded area indicating standard deviation.

according to $l(t) = \frac{1}{\max\{1, \sqrt{\log t}\}}$. We also repeat the experiment for 100 times under different seeds to generate samples.

First, we fix $\Delta t = 0.1$ and run the three learning algorithms for sufficiently long time $T = 10^6$. All the parameters to be learned are initialized as 0 except for the initial variance of the policies which is set to be 1. Figure 1 plots the running average reward trajectories, along with their standard deviations (which are all too small to be visible in the figure),

as the learning process proceeds. Among the three methods, both the PG and q-learning algorithms perform significantly better and converge to the optimal level much faster than the Q-learning. The q-learning algorithm is indeed slightly better than the PG one. Figure 2 shows the iterated values of the leaned parameters for the three algorithms. Eminently, the Q-learning has the slowest convergence. The other two algorithms, while converging at similar speed eventually, exhibit quite different behaviors on their ways to convergence in learning ψ_1 and ψ_2. The PG seems to learn these parameters faster than the q-learning at the initial stage, but then quickly overshoot the optimal level and take a long time to correct it, while the q-learning appears much smoother and does not have such an issue.

At last, we vary the Δt value to examine its impact on the performances of the three algorithms. We vary $\Delta t \in \{1, 0.1, 0.01\}$ and set $T = 10^5$ for each experiment. The learning rates and parameter initializations are fixed and the same for all the there methods. We repeat the experiment for 100 times and present the results in terms of the average running reward in Figure 3. We use the shaded area to denote the standard deviation. It is clear that the Q-learning algorithm is very sensitive to Δt. In particular, its performance worsens significantly as Δt becomes smaller. When $\Delta t = 0.01$, the algorithm almost has no improvement at all over time. This drawback of sensitivity in Δt is consistent with the observations made in Tallec et al. (2019). On the other hand, the other two algorithms show remarkable robustness across different discretization sizes.

7.3 Off-policy ergodic linear–quadratic control

The experiments reported in Subsection 7.2 are for on-policy learning. In this subsection we revisit the problem but assuming that we now have to work off-policy. Specifically, we have a sufficiently long observation of the trajectories of state, action and reward generated under a behavior policy that is not optimal. In our experiment, we take the behavior policy to be $a_t \sim \mathcal{N}(0, 1)$.

We still examine the three algorithms: "Policy Gradient" described in Algorithm 3 in Jia and Zhou (2022b), "Q-Learning" described in Appendix B, and "q-Learning" described in Algorithm 4, with different time discretization sizes. Because off-policy learning is often necessitated in cases where online interactions are costly (Uehara et al., 2022), we restrict ourselves to an *offline* dataset generated under the behavior policy. We vary $\Delta t \in \{1, 0.1, 0.01\}$ and set $T = 10^6$ for each experiment. We repeat the experiments for 100 times whose results are presented in Figure 4. Because it is generally hard or impossible to implement the learned policy and observe the resulting state and reward data in an off-policy setting, we focus on the learned *parameters* of the policy to see if they converge to the desired optimal value. The horizontal axis in Figure 4 stands for the number of iterations that are used to update the policy parameters.

In general, policy gradient methods are not applicable in the off-policy setting. This is confirmed by our experiments which show that the corresponding algorithm diverges quickly in all the cases. Q-learning still suffers from the sensitivity with respect to time discretization. The q-learning algorithm is the most stable one and converges in all scenarios. It is worth noting that in Figure 4-(a), the Q-learning and q-learning algorithms converge to the same limits, which are however different from the true optimal parameter values of the continuous-time problem. This is because the theoretical integrals involved in the value

functions are not well approximated by finite sums with coarse time discretization ($\Delta t = 1$); hence the corresponding martingale conditions under the learned parameters are not close to the theoretical continuous-time martingale conditions.
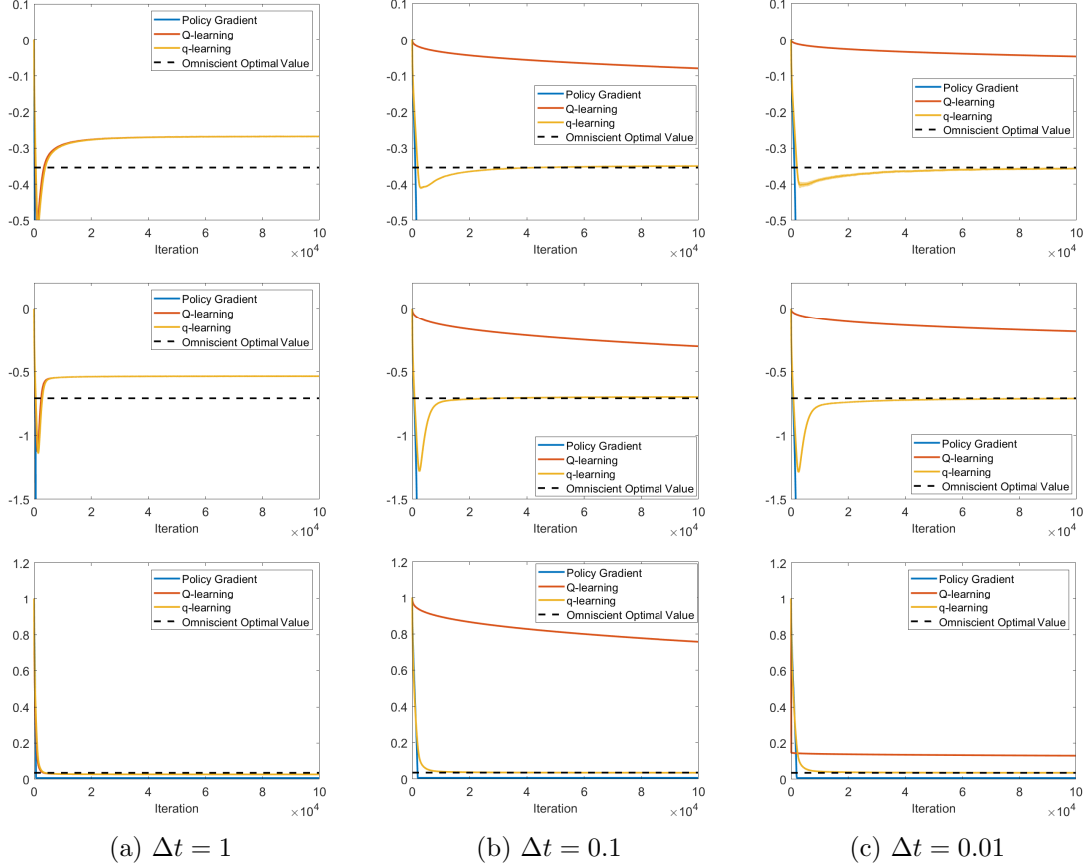
Figure 4: **Paths of learned parameters of three RL algorithms with different time discretization sizes.** A single state trajectory is generated with length $T = 10^6$ and discretized at different step sizes: $\Delta t = 1$ in (a), $\Delta t = 0.1$ in (b), and $\Delta t = 0.01$ in (c), under the behavior policy $a_t \sim \mathcal{N}(0, 1)$. From top to bottom are the paths of learned $\psi_1, \psi_2, e^{\psi_3}$ respectively. For each step size, we apply three algorithms: "Policy Gradient" described in Algorithm 3 in Jia and Zhou (2022b), "Q-Learning" described in Appendix B, and "q-Learning" described in Algorithm 4. We repeat the experiments for 100 times for each method and plot the average reward over time with the shaded area indicating standard deviation.

8 Conclusion

The previous trilogy on continuous-time RL under the entropy-regularized exploratory diffusion process framework (Wang et al., 2020; Jia and Zhou, 2022a,b) have respectively studied three key elements: optimal samplers for exploration, PE and PG. PG is a special

instance of policy improvement, and in the discrete-time MDP literature, it can be done through Q-function. However, there was no satisfying Q-learning theory in continuous time, so a different route was taken in Jia and Zhou (2022b) – to turn the PG task into a PE problem.

This paper fills this gap and adds yet another essential building block to the theoretical foundation of continuous-time RL, by developing a q-learning theory commensurate with the continuous-time setting, attacking the general policy improvement problem, and covering both on- and off-policy learning. Although the conventional Q-function provides no information about actions in continuous time, its first-order approximation, which we call the q-function, does. Moreover, it turns out that the essential component of the q-function relevant to policy updating is the Hamiltonian associated with the problem, the latter having already played a vital role in the classical model-based stochastic control theory. This fact, together with the expression of the optimal stochastic policies explicitly derived in Wang et al. (2020), in turn, explains and justifies the widely used Boltzmann exploration in classical RL for MDPs.

We characterize the q-function as the compensator to maintain the martingality of a process consisting of the value function and cumulative reward, both in the on-policy and off-policy settings. This characterization enables us to use the martingale-based PE techniques developed in Jia and Zhou (2022a) to simultaneously learn the q-function and the value function. Interestingly, the martingality is on the same process as in PE (Jia and Zhou, 2022a) but with respect to an enlarged filtration containing the policy randomization. The TD version of the resulting algorithm links to the well-known SARSA algorithm in classical Q-learning.

A significant and outstanding research question is an analysis on the convergence rates of q-learning algorithms. The existing convergence rate results for discrete-time Q-learning cannot be extended to continuous-time q-learning due to the continuous state space and general nonlinear function approximations, along with the fact that the behaviors of the Q-function and the q-function can be fundamentally different. A possible remedy is to carry out the convergence analysis within the general framework of stochastic approximation, which is poised to be the subject of a substantial future study.

We make a final observation to conclude this paper. The classical model-based approach typically separates "estimation" and "optimization", *à la* Wonham's separation theorem (Wonham, 1968) or the "plug-in" method. This approach first learns a model (including formulating and estimating its coefficients/parameters) and then optimizes over the learned model. Model-free (up to some basic dynamic structure such as diffusion processes or Markov chains) RL takes a different route: it skips estimating a model and learns optimizing policies directly via PG or Q/q-learning. The q-learning theory in this paper suggests that, to learn policies, one needs to learn the q-function or the Hamiltonian. In other words, it is the Hamiltonian which is a specific *aggregation* of the model coefficients, rather than each and every individual model coefficient, that needs to be learned/estimated for optimization. Clearly, from a pure computational standpoint, estimating a single function – the Hamiltonian – is much more efficient and robust than estimating multiple functions $(b, \sigma, r, h$ in the present paper) in terms of avoiding or reducing over-parameterization, sensitivity to errors and accumulation of errors. More importantly, however, (35) hints that the Hamiltonian/q-function can be learned through temporal differences of the value

function, so the task of learning and optimizing can be accomplished in a data-driven way. This would not be the case if we chose to learn individual model coefficients separately. This observation, we believe, highlights the fundamental differences between the model-based and model-free approaches.

Acknowledgments and Disclosure of Funding

Appendix A. Soft Q-learning in Discrete Time

We review the soft Q-learning for discrete-time Markov decision processes (MDPs) here and present the analogy to some of the major results developed in the main text. It is interesting that, to our best knowledge, the martingale perspective for MDPs is not explicitly presented. This in turn suggests that a continuous-time study may provide new aspects and insights even for discrete-time RL.

For simplicity, we consider a time-homogeneous MDP $X = \{X_t, t = 0, 1, 2, \cdots\}$ with a state space $\mathcal{X}$, an action space $\mathcal{A}$, and a transition matrix $\mathbb{P}(X_1 = x'|X_0 = x, a_0 = a) =: p(x'|x, a)$. Both $\mathcal{X}$ and $\mathcal{A}$ are finite sets. The expected reward at (x, a) is $r(x, a)$ with a discount factor $\beta \in (0, 1)$. The agent's total expected reward is $\mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t r(X_t, a_t)\right]$. A (stochastic) policy is denoted by $\boldsymbol{\pi}(\cdot|x) \in \mathcal{P}(\mathcal{A})$, which is a probability mass function on $\mathcal{A}$.

Appendix A1. Q-function associated with an arbitrary policy

Define the value function associated with a given policy $\boldsymbol{\pi}$ by

$$
\begin{aligned}
J(x; \boldsymbol{\pi}) =& \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \left[r(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}}|X_t^{\boldsymbol{\pi}})\right] \Big| X_0^{\boldsymbol{\pi}} = x\right] \\
=& \mathbb{E}\left[r(x, a_0^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_0^{\boldsymbol{\pi}}|x)\right] + \beta \mathbb{E}\left[J(X_1^{\boldsymbol{\pi}}; \boldsymbol{\pi}) \Big| X_0^{\boldsymbol{\pi}} = x\right],
\end{aligned}
\tag{45}
$$

and the Q-function associated with $\boldsymbol{\pi}$ by

$$
\begin{aligned}
Q(x, a; \boldsymbol{\pi}) =& r(x, a) + \mathbb{E}\left[\sum_{t=1}^{\infty} \beta^t \left[r(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}}|X_t^{\boldsymbol{\pi}})\right] \Big| X_0^{\boldsymbol{\pi}} = x, a_0^{\boldsymbol{\pi}} = a\right] \\
=& r(x, a) + \beta \mathbb{E}\left[J(X_1^a; \boldsymbol{\pi}) \Big| X_0^{\boldsymbol{\pi}} = x, a_0^{\boldsymbol{\pi}} = a\right].
\end{aligned}
\tag{46}
$$

Adding the entropy term $-\gamma \log \boldsymbol{\pi}(a|x)$ to both sides of (46), integrating over a and noting (45), we obtain a relation between the Q-function and the value function :

$$
\mathbb{E}\left[Q(x, a^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - J(x; \boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}(a^{\boldsymbol{\pi}}|x)\right] = 0.
\tag{47}
$$

Moreover, substituting $J(X_1^a; \boldsymbol{\pi})$ with (47) to the right hand side of (46), we further obtain

$$
Q(x, a; \boldsymbol{\pi}) = r(x, a) + \beta \mathbb{E}\left[Q(X_1^a, a_1^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}(a_1^{\boldsymbol{\pi}}|X_1^a) \Big| X_0^{\boldsymbol{\pi}} = x, a_0^{\boldsymbol{\pi}} = a\right].
\tag{48}
$$

Applying suitable algorithms (e.g., stochastic approximation) to solve the equation (48) leads to the classical SARSA algorithm.

On the other hand, rewrite (46) as

$$
J(x; \boldsymbol{\pi}) = r(x, a) - [Q(x, a; \boldsymbol{\pi}) - J(x; \boldsymbol{\pi})] + \beta \mathbb{E}\left[J(X_1^a; \boldsymbol{\pi}) \Big| X_0^{\boldsymbol{\pi}} = x, a_0^{\boldsymbol{\pi}} = a\right].
\tag{49}
$$

Recall that $A(x, a; \boldsymbol{\pi}) := Q(x, a; \boldsymbol{\pi}) - J(x; \boldsymbol{\pi})$ is called the *advantage function* for MDPs. The q-function in the main text (see (17)) is hence an advantage *rate* function in the continuous-time setting. In particular, (47) is analogous to the second constraint on q-function in (29).

Further, (49) implies that

$$M_s^{\boldsymbol{\pi}'} := \beta^s J(X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - A(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \right]$$

is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale under *any* policy $\boldsymbol{\pi}'$, where $\mathcal{F}_s$ is the σ-algebra generated by $X_0^{\boldsymbol{\pi}'}, a_0^{\boldsymbol{\pi}'}, \cdots, X_s^{\boldsymbol{\pi}'}, a_s^{\boldsymbol{\pi}'}$. To see this, we compute

$$\mathbb{E}\left[\beta^s J(X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - A(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \right] \Big| \mathcal{F}_{s-1} \right]$$

$$= \beta^s \mathbb{E}\left[J(X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \Big| X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'} \right] + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - A(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \right]$$

$$= \beta^{s-1} \left[J(X_{s-1}^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) - r(X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'}) + A(X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \right] + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - A(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \right]$$

$$= \beta^{s-1} J(X_{s-1}^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + \sum_{t=0}^{s-2} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - A(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \right].$$

$$(50)$$

Note that here $\mathcal{F}_s$ is larger than the usual "historical information set" up to time s, denoted by $\mathcal{H}_s$, which is the σ-algebra generated by $X_0^{\boldsymbol{\pi}'}, a_0^{\boldsymbol{\pi}'}, \cdots, X_s^{\boldsymbol{\pi}'}$ without observing the last $a_s^{\boldsymbol{\pi}'}$ (i.e. $\mathcal{H}_s$ is $\mathcal{F}_s$ *excluding* $a_s^{\boldsymbol{\pi}'}$). Because $M_s^{\boldsymbol{\pi}'}$ is measurable to the smaller σ-algebra $\mathcal{H}_s$, it is automatically an $(\{\mathcal{H}_s\}_{s \geq 0}, \mathbb{P})$-martingale.

The above analysis shows that $M_s^{\boldsymbol{\pi}'}$ is a martingale for *any* $\boldsymbol{\pi}'$ – whether the target policy $\boldsymbol{\pi}$ or a behavior one – so long as the advantage function A is taken as the compensator in defining $M_s^{\boldsymbol{\pi}'}$. This is the essential reason why Q-learning works for both on- and off-policy learning. However, the conclusion is not necessarily true in other types of learning methods. For example, let us take a different compensator and investigate the process $\beta^s J(X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}'} | X_t^{\boldsymbol{\pi}'}) \right]$. The continuous-time counterpart of this process is a martingale required in the policy gradient method; see (Jia and Zhou, 2022b, Theorem 4).

Now, conditioned on $\mathcal{H}_{s-1}$ and when $\boldsymbol{\pi}' = \boldsymbol{\pi}$, we have

$$\mathbb{E}\left[\beta^s J(X_s^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}} | X_t^{\boldsymbol{\pi}}) \right] \Big| \mathcal{H}_{s-1} \right]$$

$$= \beta^s \mathbb{E}\left[\mathbb{E}\left[J(X_s^{\boldsymbol{\pi}}; \boldsymbol{\pi}) | X_{s-1}^{\boldsymbol{\pi}}, a_{s-1}^{\boldsymbol{\pi}} \right] \Big| X_{s-1}^{\boldsymbol{\pi}} \right] + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}} | X_t^{\boldsymbol{\pi}}) \right]$$

$$= \beta^{s-1} \left[J(X_{s-1}^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - r(X_{s-1}^{\boldsymbol{\pi}}, a_{s-1}^{\boldsymbol{\pi}}) + A(X_{s-1}^{\boldsymbol{\pi}}, a_{s-1}^{\boldsymbol{\pi}}; \boldsymbol{\pi}) \Big| X_{s-1}^{\boldsymbol{\pi}} \right] + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}} | X_t^{\boldsymbol{\pi}}) \right]$$

$$= \beta^{s-1} J(X_{s-1}^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + \sum_{t=0}^{s-2} \beta^t \left[r(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}} | X_t^{\boldsymbol{\pi}}) \right],$$

where the last equality is due to (47), which can be applied because $a_{s-1}^{\boldsymbol{\pi}}$ is generated under the policy $\boldsymbol{\pi}$. This shows that the process is an $(\{\mathcal{H}_s\}_{s\geq 0}, \mathbb{P})$-martingale when $\boldsymbol{\pi}' = \boldsymbol{\pi}$, which in turn underpins the on-policy learning.

However, when conditioned on $\mathcal{F}_{s-1}$, we have

$$\mathbb{E}\left[\beta^s J(X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}'}|X_t^{\boldsymbol{\pi}'})\right] \Big| \mathcal{F}_{s-1}\right]$$

$$= \beta^s \mathbb{E}\left[J(X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \Big| X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'}\right] + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}'}|X_t^{\boldsymbol{\pi}'})\right]$$

$$= \beta^{s-1}\left[J(X_{s-1}^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) - r(X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'}) + A(X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'}; \boldsymbol{\pi})\right] + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}'}|X_t^{\boldsymbol{\pi}'})\right]$$

$$= \beta^{s-1} J(X_{s-1}^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + \sum_{t=0}^{s-2} \beta^t \left[r(X_t^{\boldsymbol{\pi}'}, a_t^{\boldsymbol{\pi}'}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}'}|X_t^{\boldsymbol{\pi}'})\right]$$

$$+ \beta^{s-1}\left[A(X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}'}|X_t^{\boldsymbol{\pi}'})\right].$$

So the same process is *not* an $(\{\mathcal{F}_s\}_{s\geq 0}, \mathbb{P})$-martingale in general, unless $A(X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) = \gamma \log \boldsymbol{\pi}(a_t^{\boldsymbol{\pi}'}|X_t^{\boldsymbol{\pi}'})$ for any $X_{s-1}^{\boldsymbol{\pi}'}, a_{s-1}^{\boldsymbol{\pi}'}$. Two conclusions can be drawn from this example: on one hand, policy gradient works for on-policy but not off-policy, and on the other hand, it is important to choose the correct filtration to ensure martingality and hence the correct test functions in designing learning algorithms.

The analysis in (50) yields a joint martingale characterization of (J, A), analogous to Theorem 7. It is curious that, to our best knowledge, such a martingale condition, while not hard to derive in the MDP setting, has not been explicitly introduced nor utilized to study Q-learning in the existing RL literature.

Appendix A2. Q-function associated with the optimal policy

Now consider the value function and Q-function that are associated with the optimal policy, denoted by J^* and Q^* respectively. Recall that the Bellman equation implies

$$J^*(x) = \sup_{\boldsymbol{\pi}} \mathbb{E}_{a\sim\boldsymbol{\pi}} \left\{r(x, a) - \gamma \log \boldsymbol{\pi}(a|x) + \beta \mathbb{E}\left[J^*(X_1^a) \Big| X_0 = x, a_0 = a\right]\right\}. \qquad (51)$$

It follows from (46) that

$$Q^*(x, a) = r(x, a) + \beta \mathbb{E}\left[J^*(X_1^a) \Big| X_0 = x, a_0 = a\right]. \qquad (52)$$

Hence (51) becomes

$$J^*(x) = \sup_{\boldsymbol{\pi}} \mathbb{E}_{a\sim\boldsymbol{\pi}} \left[Q^*(x, a) - \gamma \log \boldsymbol{\pi}(a|x)\right].$$

Solving the optimization on the right hand side of the above we get the optimal policy $\boldsymbol{\pi}^*(a|x) \propto \exp\{\frac{1}{\gamma} Q^*(x, a)\}$ or $\boldsymbol{\pi}^*(a|x) = \frac{\exp\{\frac{1}{\gamma} Q^*(x,a)\}}{\sum_{a\in\mathcal{A}} \exp\{\frac{1}{\gamma} Q^*(x,a)\}}$. Denote

$$\mathrm{soft}_\gamma \max_a Q^*(x, a) := \mathbb{E}_{a\sim\boldsymbol{\pi}^*} \left[Q^*(x, a) - \gamma \log \boldsymbol{\pi}^*(a|x)\right] \equiv \gamma \log \sum_{a\in\mathcal{A}} \exp\{\frac{1}{\gamma} Q^*(x, a)\}.$$

Then the Bellman equation becomes $J^*(x) = \text{soft}_\gamma \max_a Q^*(x,a)$ and (52) reduces to

$$Q^*(x,a) = r(x,a) + \beta \mathbb{E}\left[\text{soft}_\gamma \max_{a'} Q^*(X_1^a, a') \Big| X_0 = x, a_0 = a\right], \tag{53}$$

which is the foundation for (off-policy) Q-learning algorithms; see e.g. (Sutton and Barto, 2018, p. 131).

On the other hand, because $J^*(x) = \text{soft}_\gamma \max_a Q^*(x,a) = \gamma \log \sum_{a \in \mathcal{A}} \exp\{\frac{1}{\gamma} Q^*(x,a)\}$, we have

$$\sum_{a \in \mathcal{A}} \exp\left\{\frac{1}{\gamma}[Q^*(x,a) - J^*(x)]\right\} = 1. \tag{54}$$

Recall the advantage function $A^*(x,a) = Q^*(x,a) - J^*(x)$. Thus (54) is analogous to (25). Moreover, rearranging (52) yields

$$J^*(x) = r(x,a) - [Q^*(x,a) - J^*(x)] + \beta \mathbb{E}\left[J^*(X_1^a) \Big| X_0 = x, a_0 = a\right].$$

Based on a similar derivation to that in Appendix A1, we obtain

$$\beta^s J^*(X_s^{\boldsymbol{\pi}}) + \sum_{t=0}^{s-1} \beta^t \left[r(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}) - A^*(X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}})\right]$$

is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale for any policy $\boldsymbol{\pi}$; hence it is analogous to the martingale characterization of optimal q-function in Theorem 9.

Appendix B. Δt-Parameterized Q-Learning

Instead of the (little) q-learning approach developed in this paper, one can also apply the conventional (big) Q-learning method to the Q-function $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$ defined in (15). Note that $\Delta t > 0$ becomes a *parameter* in the latter approach. In this Appendix, we review this Δt-parameterized Q-learning and the associated Q-learning algorithms, which are used in the simulation experiments of the paper for comparison purpose.

Given a policy $\boldsymbol{\pi}$ and a time step $\Delta t > 0$, Q-learning focuses on learning $Q_{\Delta t}(t, x, a; \boldsymbol{\pi})$. As with q-learning, there are two distinctive scenarios: when the normalizing constant is easily available and when it is not. In the following, we denote by $Q_{\Delta t}^\psi$ the function approximator to the Q-function where $\psi \in \Psi \subset \mathbb{R}^{L_\psi}$ is the finite dimensional parameter vector to be learned, and by $\boldsymbol{\pi}^\psi$ the associated policy where $\boldsymbol{\pi}^\psi(a|t,x) \propto \exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(t, x, a)\}$.

Appendix B1. When normalizing constant is available

When the normalization constant $\int_{\mathcal{A}} \exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(t, x, a)\} \mathrm{d}a$ can be explicitly calculated, we can obtain the exact expression of $\boldsymbol{\pi}^\psi$ as

$$\boldsymbol{\pi}^\psi(a|t,x) = \frac{\exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(t, x, a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(t, x, a)\} \mathrm{d}a}.$$

Then, the conventional Q-learning method (e.g, SARSA, Sutton and Barto 2018, Chapter 10) leads to the following updating rule of the Q-function parameters ψ:

$$
\begin{aligned}
\psi \leftarrow \psi + \alpha_\psi \bigg(& Q_{\Delta t}^\psi(t + \Delta t, X_{t+\Delta t}^{a_t}, a_{t+\Delta t}) - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}|t + \Delta t, X_{t+\Delta t}^{a_t})\Delta t \\
& - Q_{\Delta t}^\psi(t, X_t, a_t) + r(t, X_t, a_t)\Delta t - \beta Q_{\Delta t}^\psi(t, X_t, a_t)\Delta t \bigg) \frac{\partial Q_{\Delta t}^\psi}{\partial \psi}(t, X_t, a_t),
\end{aligned}
\tag{55}
$$

where $a_t \sim \boldsymbol{\pi}^\psi(\cdot|t, X_t)$ and $a_{t+\Delta t} \sim \boldsymbol{\pi}^\psi(\cdot|t + \Delta t, X_{t+\Delta t})$.

Appendix B2. When normalizing constant is unavailable

When the normalizing constant is not available, we cannot compute the exact expression of $\boldsymbol{\pi}^\psi$. In this case we follow an analysis analogous to that in Subsection 5.1 by using a family of policies $\{\boldsymbol{\pi}^\phi(a|t, x)\}_{\phi \in \Phi}$ whose densities can be easily calculated to approximate $\boldsymbol{\pi}^\psi$.

Theorem 10 implies that if we start from a policy $\boldsymbol{\pi}^\phi$, $\phi \in \Phi$, and evaluate its corresponding Q-function $\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(t, x, a) \approx \frac{1}{\gamma} H\big(t, x, \cdot, \frac{\partial J}{\partial x}(t, x; \boldsymbol{\pi}^\phi), \frac{\partial^2 J}{\partial x^2}(t, x; \boldsymbol{\pi}^\phi)\big) + \frac{1}{\gamma \Delta t} J(t, x; \boldsymbol{\pi}^\phi)$, then a new policy $\boldsymbol{\pi}^{\phi'}$, $\phi' \in \Phi$, improves $\boldsymbol{\pi}^\phi$ so long as

$$
D_{KL}\left(\boldsymbol{\pi}^{\phi'}(\cdot|t, x) \Big\| \exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(t, x, \cdot)\} \right) \leq D_{KL}\left(\boldsymbol{\pi}^\phi(\cdot|t, x) \Big\| \exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(t, x, \cdot)\} \right).
$$

Thus, following the same lines of argument as in Subsection 5.1, we derive the updating rule of ϕ every step as

$$
\phi \leftarrow \phi - \alpha_\phi \left[\log \boldsymbol{\pi}^\phi(a_t|t, X_t) - \frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(t, X_t, a_t) \right] \frac{\partial}{\partial \phi} \log \boldsymbol{\pi}^\phi(a_t|t, X_t),
$$

where $a_t \sim \boldsymbol{\pi}^\phi(\cdot|t, X_t)$.

Appendix B3. Ergodic tasks

For the general regularized ergodic problem formulated in Section 6, we define the Q-function:

$$
Q_{\Delta t}(x, a; \boldsymbol{\pi}) = J(x; \boldsymbol{\pi}) + \left[H\big(x, a, \frac{\partial J}{\partial x}(x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(x; \boldsymbol{\pi})\big) - V(\boldsymbol{\pi}) \right]\Delta t - \bar{J}.
\tag{56}
$$

Then we have all the parallel results regarding relation among the Q-function, the value function and the intertemporal increment of the Q-function, and devise algorithms accordingly.

For illustration, we only present the results when

$$
\boldsymbol{\pi}^\psi(a|x) = \frac{\exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(x, a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^\psi(x, a)\}\mathrm{d}a}
$$

48

is explicitly available. The updating rules of the parameters ψ and V are

$$
\begin{aligned}
\psi \leftarrow \psi + \alpha_\psi \Bigg(& Q_{\Delta t}^\psi(X_{t+\Delta t}^{a_t}, a_{t+\Delta t}) - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}|X_{t+\Delta t}^{a_t})\Delta t \\
& - Q_{\Delta t}^\psi(X_t, a_t) + r(X_t, a_t)\Delta t - V\Delta t \Bigg) \frac{\partial Q_{\Delta t}^\psi}{\partial \psi}(X_t, a_t) \\
V \leftarrow V + \alpha_V \Bigg(& Q_{\Delta t}^\psi(X_{t+\Delta t}^{a_t}, a_{t+\Delta t}) - \gamma \log \boldsymbol{\pi}^\psi(a_{t+\Delta t}|X_{t+\Delta t}^{a_t})\Delta t \\
& - Q_{\Delta t}^\psi(X_t, a_t) + r(X_t, a_t)\Delta t - V\Delta t \Bigg),
\end{aligned}
\tag{57}
$$

where $a_t \sim \boldsymbol{\pi}^\psi(\cdot|X_t)$ and $a_{t+\Delta t} \sim \boldsymbol{\pi}^\psi(\cdot|X_{t+\Delta t})$.

Appendix B4. Q-learning for mean–variance portfolio selection

Motivated by the form of the solution in Wang and Zhou (2020), we parameterize the Q-function by

$$
Q_{\Delta t}^\psi(t, x, a; w) = -e^{-\psi_3(T-t)}\left[(x-w)^2 + \psi_2 a(x-w) + \frac{1}{2}e^{-\psi_1}a^2\right] + \psi_4(t^2 - T^2) + \psi_5(t-T) + (w-z)^2.
$$

Here we use $-e^{-\psi_1} < 0$ to guarantee the concavity of the Q-function in a so that the function can be maximized.

This Q-function in turn gives rise to an explicit form of policy, which is Gaussian:

$$
\boldsymbol{\pi}^\psi(\cdot|t, x; w) = \mathcal{N}\left(-\psi_2 e^{\psi_1}(x-w), \gamma \Delta t e^{\psi_3(T-t)+\psi_1}\right) \propto \exp\{\frac{1}{\gamma \Delta t}Q_{\Delta t}^\psi(t, x, \cdot)\}.
$$

We need the following derivative in the corresponding Q-learning algorithms:

$$
\frac{\partial Q_{\Delta t}^\psi}{\partial \psi}(t, x, a; w) = \begin{pmatrix} \frac{1}{2}e^{-\psi_3(T-t)}e^{-\psi_1}a^2 \\ -e^{-\psi_3(T-t)}e^{-\psi_1}a(x-w) \\ (T-t)e^{-\psi_3(T-t)}[(x-w)^2 + \psi_2 a(x-w) + \frac{1}{2}e^{-\psi_1}a^2] \\ t^2 - T^2 \\ t - T \end{pmatrix}.
$$

The parameters in the Q-function can then be updated based on (55) and the Lagrange multiplier can be updated similarly as in Algorithm 5.

Appendix B5. Q-learning for ergodic LQ control

For the ergodic LQ control problem formulated in Subsection 7.2, we can prove that the value function and the Hamiltonian are both quadratic in (x, a); see, e.g., Wang et al. (2020). Therefore we parameterize the Q-function by a general quadratic function:

$$
Q_{\Delta t}^\psi(x, a) = -\frac{e^{-\psi_3}}{2}(a - \psi_1 x - \psi_2)^2 + \psi_4 x^2 + \psi_5 x.
$$

Here we omit the constant term since the value function is unique only up to a constant. We also use $-e^{-\psi_3} < 0$ to ensure that the Q-function is concave in a. The optimal value V is an extra parameter.

The Q-function leads to

$$\boldsymbol{\pi}^{\psi}(\cdot|x) = \mathcal{N}\left(\psi_1 x + \psi_2, \gamma \Delta t e^{\psi_3}\right) \propto \exp\{\frac{1}{\gamma \Delta t} Q_{\Delta t}^{\psi}(x, \cdot)\}.$$

Finally, we compute the following derivative for use in Q-learning algorithms:

$$\frac{\partial Q_{\Delta t}^{\psi}}{\partial \psi}(x, a) = \left(e^{-\psi_3}(a - \psi_1 x - \psi_2)x, e^{-\psi_3}(a - \psi_1 x - \psi_2), \frac{e^{-\psi_3}}{2}(a - \psi_1 x - \psi_2)^2, x^2, x\right)^{\top}.$$

Then the parameters can be updated based on (57).

Appendix C. Proofs of Statements

Proof of Theorem 2

We first prove a preliminary result about the entropy maximizing density.

Lemma 13 *Let $\gamma > 0$ and a measurable function $q : \mathcal{A} \to \mathbb{R}$ with $\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}q(a)\}da < \infty$ be given. Then $\pi^*(a) = \frac{\exp\{\frac{1}{\gamma}q(a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}q(a)\}da} \in \mathcal{P}(\mathcal{A})$ is the unique maximizer of the following problem*

$$\max_{\pi(\cdot) \in \mathcal{P}(\mathcal{A})} \int_{\mathcal{A}} [q(a) - \gamma \log \pi(a)]\pi(a)da. \tag{58}$$

Proof Set $\nu = 1 - \log \int_{\mathcal{A}} \exp\{\frac{1}{\gamma}q(a)\}da$ and consider a new problem:

$$\max_{\pi(\cdot) \in \mathcal{P}(\mathcal{A})} \int_{\mathcal{A}} \left\{[q(a) - \gamma \log \pi(a)]\pi(a) + \gamma\nu\pi(a)\right\}da - \gamma\nu. \tag{59}$$

Noting $\int_{\mathcal{A}} \pi(a)da = 1$ due to $\pi(\cdot) \in \mathcal{P}(\mathcal{A})$, problems (58) and (59) are equivalent; hence we focus on problem (59). Relaxing the constraint of being a density function we deduce

$$\max_{\pi(\cdot) \in \mathcal{P}(\mathcal{A})} \int_{\mathcal{A}} \left\{[q(a) - \gamma \log \pi(a)]\pi(a) + \gamma\nu\pi(a)\right\}da - \gamma\nu$$

$$\leq \max_{\pi(\cdot) > 0} \int_{\mathcal{A}} \left\{[q(a) - \gamma \log \pi(a)]\pi(a) + \gamma\nu\pi(a)\right\}da - \gamma\nu \tag{60}$$

$$\leq \int_{\mathcal{A}} \max_{\pi > 0} \left\{[q(a) - \gamma \log \pi]\pi + \gamma\nu\pi\right\}da - \gamma\nu.$$

The unique maximizer of the inner optimization on the right hand side of the above is

$$\exp\{\frac{1}{\gamma}q(a) + \nu - 1\} = \frac{\exp\{\frac{1}{\gamma}q(a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}q(a)\}da} =: \pi^*(a).$$

As $\pi^* \in \mathcal{P}(\mathcal{A})$, it is the optimal solution to (59) or, equivalently, (58). The uniqueness follows from that of the inner optimization in (60). ∎

Now we are ready to prove Theorem 2. To ease notation we use the q-function q even though it had not been introduced prior to Theorem 2 in the main text. We also employ the results of Theorem 6 whose proof does not rely on Theorem 2 nor its proof here.

Recall from (16), $q(t, x, a; \boldsymbol{\pi})$ is defined to be

$$q(t, x, a; \boldsymbol{\pi}) = \frac{\partial J}{\partial t}(t, x; \boldsymbol{\pi}) + H\left(t, x, a, \frac{\partial J}{\partial x}(t, x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t, x; \boldsymbol{\pi})\right) - \beta J(t, x; \boldsymbol{\pi}),$$

with the Hamiltonian H defined in (3).

Proof [Proof of Theorem 2]

We work on the equivalent formulation (8)–(9). For two given admissible policies $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$, and any $0 \leq t \leq T$, apply Itô's lemma to the process $e^{-\beta s} J(s, X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi})$ from $s \in [t, T]$, which is the value function under $\boldsymbol{\pi}$ but over the state process under $\boldsymbol{\pi}'$:

$$e^{-\beta T} J(T, \tilde{X}_T^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) - e^{-\beta t} J(t, \tilde{X}_t^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + \int_t^T e^{-\beta s} \int_{\mathcal{A}} [r(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a) - \gamma \log \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'})] \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s$$

$$= \int_t^T e^{-\beta s} \int_{\mathcal{A}} \big[\frac{\partial J}{\partial t}(s, \tilde{X}_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + H\big(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a, \frac{\partial J}{\partial x}(s, \tilde{X}_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(s, \tilde{X}_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi})\big) - \beta J(s, \tilde{X}_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi})$$

$$- \gamma \log \boldsymbol{\pi}'(a|s, X_s^{\boldsymbol{\pi}'})\big] \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s + \int_t^T e^{-\beta s} \frac{\partial}{\partial x} J(s, \tilde{X}_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \circ \tilde{\sigma}\big(s, \tilde{X}_s^{\boldsymbol{\pi}'}, \boldsymbol{\pi}'(\cdot|s, \tilde{X}_s^{\boldsymbol{\pi}'})\big) \mathrm{d}W_s$$

$$= \int_t^T e^{-\beta s} \int_{\mathcal{A}} \big[q(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a; \boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}'(a|s, X_s^{\boldsymbol{\pi}'}) \big] \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s$$

$$+ \int_t^T e^{-\beta s} \frac{\partial}{\partial x} J(s, \tilde{X}_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \circ \tilde{\sigma}\big(s, \tilde{X}_s^{\boldsymbol{\pi}'}, \boldsymbol{\pi}'(\cdot|s, \tilde{X}_s^{\boldsymbol{\pi}'})\big) \mathrm{d}W_s.$$

Because $\boldsymbol{\pi}' = \mathcal{I}\boldsymbol{\pi}$, it follows from Lemma 13 that for any $(s, y) \in [0, T] \times \mathbb{R}^d$, we have

$$\int_{\mathcal{A}} \big[q(s, y, a; \boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}'(a|s, y) \big] \boldsymbol{\pi}'(a|s, y) \mathrm{d}a \geq \int_{\mathcal{A}} \big[q(s, y, a; \boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}(a|s, y) \big] \boldsymbol{\pi}(a|s, y) \mathrm{d}a = 0,$$

where the equality is due to (20) in Theorem 6. Thus,

$$e^{-\beta T} J(T, \tilde{X}_T^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) - e^{-\beta t} J(t, \tilde{X}_t^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) + \int_t^T e^{-\beta s} \int_{\mathcal{A}} [r(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a) - \gamma \log \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'})] \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s$$

$$\geq \int_t^T e^{-\beta s} \frac{\partial}{\partial x} J(s, \tilde{X}_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \circ \tilde{\sigma}\big(s, \tilde{X}_s^{\boldsymbol{\pi}'}, \boldsymbol{\pi}'(\cdot|s, \tilde{X}_s^{\boldsymbol{\pi}'})\big) \mathrm{d}W_s.$$

The above argument and the resulting inequalities are also valid when T is replaced by $T \wedge u_n$, where $u_n = \inf\{s \geq t : |\tilde{X}_s^{\boldsymbol{\pi}'}| \geq n\}$ is a sequence of stopping times. Therefore,

$$J(t, x; \boldsymbol{\pi}) \leq \mathbb{E}^{\mathbb{P}^W} \bigg[e^{-\beta(T \wedge u_n - t)} h(\tilde{X}_{T \wedge u_n}^{\boldsymbol{\pi}'})$$

$$+ \int_t^{T \wedge u_n} e^{-\beta(s-t)} \int_{\mathcal{A}} [r(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a) - \gamma \log \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'})] \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \bigg].$$

$$(61)$$

It follows from Assumption 1-(iv), Definition 1-(iii) and the moment estimate in Jia and Zhou (2022b, Lemma 2) that there exist constants $\mu, C > 0$ such that

$$\left| \mathbb{E}^{\mathbb{P}^W} \left[\int_t^{T \wedge u_n} e^{-\beta(s-t)} \int_{\mathcal{A}} [r(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a) - \gamma \log \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'})] \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right] \right|$$

$$\leq \mathbb{E}^{\mathbb{P}^W} \left[\int_t^T \int_{\mathcal{A}} |r(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a) - \gamma \log \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'})| \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right]$$

$$\leq C \int_t^T \mathbb{E}^{\mathbb{P}^W} \left[(1 + |\tilde{X}_s^{\boldsymbol{\pi}'}|^\mu) \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right] \mathrm{d}t \leq C(1 + |x|^\mu).$$

By the dominated convergence theorem, we have as $n \to \infty$,

$$\mathbb{E}^{\mathbb{P}^W} \left[\int_t^{T \wedge u_n} e^{-\beta(s-t)} \int_{\mathcal{A}} [r(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a) - \gamma \log \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'})] \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right]$$

$$\to \mathbb{E}^{\mathbb{P}^W} \left[\int_t^T e^{-\beta(s-t)} \int_{\mathcal{A}} [r(s, \tilde{X}_s^{\boldsymbol{\pi}'}, a) - \gamma \log \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'})] \boldsymbol{\pi}'(a|s, \tilde{X}_s^{\boldsymbol{\pi}'}) \mathrm{d}a \mathrm{d}s \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right].$$

In addition,

$$\mathbb{E}^{\mathbb{P}^W} \left[e^{-\beta(T \wedge u_n - t)} h(\tilde{X}_{T \wedge u_n}^{\boldsymbol{\pi}'}) \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right]$$

$$\leq \mathbb{E}^{\mathbb{P}^W} \left[h(\tilde{X}_T^{\boldsymbol{\pi}'}) \mathbb{1}_{\{\max_{t \leq s \leq T} |\tilde{X}_T^{\boldsymbol{\pi}'}| < n\}} \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right] + \mathbb{E}^{\mathbb{P}^W} \left[h(\tilde{X}_{u_n}^{\boldsymbol{\pi}'}) \mathbb{1}_{\{\max_{t \leq s \leq T} |\tilde{X}_T^{\boldsymbol{\pi}'}| \geq n\}} \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right]$$

$$\leq C \mathbb{E}^{\mathbb{P}^W} \left[(1 + |\tilde{X}_T^{\boldsymbol{\pi}'}|^\mu) \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right] + C(1 + n^\mu) \mathbb{E}^{\mathbb{P}^W} \left[\mathbb{1}_{\{\max_{t \leq s \leq T} |\tilde{X}_T^{\boldsymbol{\pi}'}| \geq n\}} \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right]$$

$$\leq C(1 + |x|^\mu) + C(1 + n^\mu) \frac{\mathbb{E}^{\mathbb{P}^W} \left[\max_{t \leq s \leq T} |\tilde{X}_T^{\boldsymbol{\pi}'}|^{\mu+1} \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right]}{n^{\mu+1}}$$

$$\leq C(1 + |x|^\mu) + \frac{C(1 + n^\mu)(1 + |x|^{\mu+1})}{n^{\mu+1}} < \infty.$$

Again, by the dominated convergence theorem, we have as $n \to \infty$,

$$\mathbb{E}^{\mathbb{P}^W} \left[e^{-\beta(T \wedge u_n - t)} h(\tilde{X}_{T \wedge u_n}^{\boldsymbol{\pi}'}) \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right] \to \mathbb{E}^{\mathbb{P}^W} \left[e^{-\beta(T-t)} h(\tilde{X}_T^{\boldsymbol{\pi}'}) \Big| \tilde{X}_t^{\boldsymbol{\pi}'} = x \right].$$

Hence, sending $n \to \infty$, we conclude from (61) that $J(t, x; \boldsymbol{\pi}) \leq J(t, x; \boldsymbol{\pi}')$. This proves that $\boldsymbol{\pi}' = \mathcal{I}\boldsymbol{\pi}$ improves upon $\boldsymbol{\pi}$. Moreover, if $\mathcal{I}\boldsymbol{\pi} \equiv \boldsymbol{\pi}' = \boldsymbol{\pi}$, then $J(t, x; \boldsymbol{\pi}) \equiv J(t, x; \boldsymbol{\pi}') =: J^*(t, x)$, which satisfies the PDE (11). However, Lemma 13 shows that

$$\int_{\mathcal{A}} \left[H\left(t, x, a, \frac{\partial J^*}{\partial x}(t, x), \frac{\partial^2 J^*}{\partial x^2}(t, x)\right) - \gamma \log \boldsymbol{\pi}'(a|t, x) \right] \boldsymbol{\pi}'(a|t, x) \mathrm{d}a$$

$$= \sup_{\boldsymbol{\pi} \in \mathcal{P}(\mathcal{A})} \int_{\mathcal{A}} \left[H\left(t, x, a, \frac{\partial J^*}{\partial x}(t, x), \frac{\partial^2 J^*}{\partial x^2}(t, x)\right) - \gamma \log \boldsymbol{\pi}(a) \right] \boldsymbol{\pi}(a) \mathrm{d}a.$$

This means that J^* also satisfies the HJB equation (12), implying that J^* is the optimal value function and hence $\boldsymbol{\pi}$ is the optimal policy. ∎

Proof of Proposition 3

Proof Breaking the full time period $[t, T]$ into subperiods $[t, t + \Delta t)$ and $[t + \Delta t, T]$, while conditioning on the state at $t + \Delta t$ for the second subperiod, we obtain:

$$
\begin{aligned}
& Q_{\Delta t}(t, x, a; \boldsymbol{\pi}) \\
=& \mathbb{E}^{\mathbb{P}}\Big[\int_t^{t+\Delta t} e^{-\beta(s-t)} r(s, X_s^a, a) \mathrm{d}s \\
& + \mathbb{E}^{\mathbb{P}}\big[\int_{t+\Delta t}^T e^{-\beta(s-t)}[r(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_s^{\boldsymbol{\pi}}|s, X_s^{\boldsymbol{\pi}})]\mathrm{d}s + e^{-\beta(T-t)} h(X_T^{\boldsymbol{\pi}}) | X_{t+\Delta t}^a\big] \Big| X_t^{\tilde{\boldsymbol{\pi}}} = x\Big] \\
=& \mathbb{E}^{\mathbb{P}^W}\Big[\int_t^{t+\Delta t} e^{-\beta(s-t)} r(s, X_s^a, a)\mathrm{d}s + e^{-\beta \Delta t} J(t + \Delta t, X_{t+\Delta t}^a; \boldsymbol{\pi}) \Big| X_t^{\tilde{\boldsymbol{\pi}}} = x\Big] \\
=& \mathbb{E}^{\mathbb{P}^W}\Big[\int_t^{t+\Delta t} e^{-\beta(s-t)} r(s, X_s^a, a)\mathrm{d}s + e^{-\beta \Delta t} J(t + \Delta t, X_{t+\Delta t}^a; \boldsymbol{\pi}) - J(t, x; \boldsymbol{\pi}) \Big| X_t^{\tilde{\boldsymbol{\pi}}} = x\Big] + J(t, x; \boldsymbol{\pi}) \\
=& \mathbb{E}^{\mathbb{P}^W}\Big[\int_t^{t+\Delta t} e^{-\beta(s-t)} \big[\frac{\partial J}{\partial t}(s, X_s^a; \boldsymbol{\pi}) + H\big(s, X_s^a, a, \frac{\partial J}{\partial x}(s, X_s^a; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(s, X_s^a; \boldsymbol{\pi})\big) \\
& - \beta J(s, X_s^a; \boldsymbol{\pi})\big]\mathrm{d}s \Big| X_t^{\tilde{\boldsymbol{\pi}}} = x\Big] + J(t, x; \boldsymbol{\pi}) \\
=& J(t, x; \boldsymbol{\pi}) + \Big[\frac{\partial J}{\partial t}(t, x; \boldsymbol{\pi}) + H\big(t, x, a, \frac{\partial J}{\partial x}(t, x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t, x; \boldsymbol{\pi})\big) - \beta J(t, x; \boldsymbol{\pi})\Big] \Delta t + o(\Delta t),
\end{aligned}
$$
(62)

where the second to the last equality follows from Itô's lemma and the last equality and the error order are due to the approximation of the integral involved. ∎

Proof of Corollary 5

Proof Its proof directly follows from the results of Proposition 3. ∎

Proof of Theorem 6

Proof First, (20) follows readily from its definition in Definition 4, the Feynman–Kac formula (11), and the fact that $\frac{\partial J}{\partial t}(t, x; \boldsymbol{\pi})$ and $\beta J(t, x; \boldsymbol{\pi})$ both do not depend on action a.

(i) We now focus on (18). Applying Itô's lemma to the process $e^{-\beta s}J(s, X_s^{\boldsymbol{\pi}}; \boldsymbol{\pi})$, we obtain for $0 \le t < s \le T$:

$$e^{-\beta s}J(s, X_s^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - e^{-\beta t}J(t, x; \boldsymbol{\pi}) + \int_t^s e^{-\beta u}[r(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}) - \hat{q}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})]\mathrm{d}u$$

$$= \int_t^s e^{-\beta u}\Big[\frac{\partial J}{\partial t}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}) + H\big(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}, \frac{\partial J}{\partial x}(u, X_u^{\boldsymbol{\pi}}; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(u, X_u^{\boldsymbol{\pi}}; \boldsymbol{\pi})\big) - \beta J(u, X_u^{\boldsymbol{\pi}}; \boldsymbol{\pi})$$

$$- \hat{q}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})\Big]\mathrm{d}u + \int_t^s e^{-\beta u}\frac{\partial}{\partial x}J(u, X_u^{\boldsymbol{\pi}}; \boldsymbol{\pi}) \circ \mathrm{d}W_u$$

$$= \int_t^s e^{-\beta u}\big[q(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - \hat{q}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})\big]\mathrm{d}u + \int_t^s e^{-\beta u}\frac{\partial}{\partial x}J(u, X_u^{\boldsymbol{\pi}}; \boldsymbol{\pi}) \circ \sigma(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})\mathrm{d}W_u.$$

Recall that $\{a_s^{\boldsymbol{\pi}}, t \le s \le T\}$ is $\{\mathcal{F}_s\}_{s \ge 0}$-progressively measurable. So, if $\hat{q} \equiv q(\cdot, \cdot, \cdot; \boldsymbol{\pi})$, then the above process, and hence (18), is an $(\{\mathcal{F}_s\}_{s \ge 0}, \mathbb{P})$-martingale on $[t, T]$.

Conversely, if the right hand side of the above is a martingale, then, because its second term is a local martingale, we have that $\int_t^s e^{-\beta u}\big[q(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - \hat{q}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})\big]\mathrm{d}u$ is a continuous local martingale with finite variation and hence zero quadratic variation. Therefore, $\mathbb{P}$-almost surely, $\int_t^s e^{-\beta u}\big[q(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}; \boldsymbol{\pi}) - \hat{q}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})\big]\mathrm{d}u = 0$ for all $s \in [t, T]$; see, e.g., (Karatzas and Shreve, 2014, Chapter 1, Exercise 5.21).

Denote $f(t, x, a) = q(t, x, a; \boldsymbol{\pi}) - \hat{q}(t, x, a)$. Then f is a continuous function that maps $[0, T] \times \mathbb{R}^d \times \mathcal{A}$ to $\mathbb{R}$. Suppose the desired conclusion is not true, then there exists a triple (t^*, x^*, a^*) and $\epsilon > 0$ such that $f(t^*, x^*, a^*) > \epsilon$. Because f is continuous, there exists $\delta > 0$ such that $f(u, x', a') > \epsilon/2$ for all (u, x', a') with $|u - t^*| \vee |x' - x^*| \vee |a' - a^*| < \delta$. Here "$\vee$" means taking the larger one, i.e., $u \vee v = \max\{u, v\}$.

Now consider the state process, still denoted by $X^{\boldsymbol{\pi}}$, starting from (t^*, x^*, a^*), namely, $\{X_s^{\boldsymbol{\pi}}, t^* \le s \le T\}$ follows (6) with $X_{t^*}^{\boldsymbol{\pi}} = x^*$ and $a_{t^*}^{\boldsymbol{\pi}} = a^*$. Define

$$\tau = \inf\{u \ge t^* : |u - t^*| > \delta \text{ or } |X_u^{\boldsymbol{\pi}} - x^*| > \delta\} = \inf\{u \ge t^* : |X_u^{\boldsymbol{\pi}} - x^*| > \delta\} \wedge (t^* + \delta).$$

The continuity of $X^{\boldsymbol{\pi}}$ implies that $u > t^*$, $\mathbb{P}$-almost surely. Here "$\wedge$" means taking the smaller one, i.e., $u \wedge v = \min\{u, v\}$.

We have already proved that there exists $\Omega_0 \in \mathcal{F}$ with $\mathbb{P}(\Omega_0) = 0$ such that for all $\omega \in \Omega \setminus \Omega_0$, $\int_{t^*}^s e^{-\beta u}f(u, X_u^{\boldsymbol{\pi}}(\omega), a_u^{\boldsymbol{\pi}}(\omega))\mathrm{d}u = 0$ for all $s \in [t^*, T]$. It follows from Lebesgue's differentiation theorem that for any $\omega \in \Omega \setminus \Omega_0$,

$$f(s, X_s^{\boldsymbol{\pi}}(\omega), a_s^{\boldsymbol{\pi}}(\omega)) = 0, \text{ a.e. } s \in [t^*, \tau(\omega)].$$

Consider the set $Z(\omega) = \{s \in [t^*, \tau(\omega)] : a_s^{\boldsymbol{\pi}}(\omega) \in \mathcal{B}_\delta(a^*)\} \subset [t^*, \tau(\omega)]$, where $\mathcal{B}_\delta(a^*) = \{a' \in \mathcal{A} : |a' - a^*| > \delta\}$ is the neighborhood of a^*. Because $f(s, X_s^{\boldsymbol{\pi}}(\omega), a_s^{\boldsymbol{\pi}}(\omega)) > \frac{\epsilon}{2}$ when $s \in Z(\omega)$, we conclude that $Z(\omega)$ has Lebesgue measure zero for any $\omega \in \Omega \setminus \Omega_0$. That is,

$$\int_{[t^*, T]} \mathbb{1}_{\{s \le \tau(\omega)\}} \mathbb{1}_{\{a_s^{\boldsymbol{\pi}}(\omega) \in \mathcal{B}_\delta(a^*)\}}\mathrm{d}s = 0.$$

Integrating ω with respect to $\mathbb{P}$ and applying Fubini's theorem, we obtain

$$
0 = \int_\Omega \int_{[t^*,T]} \mathbb{1}_{\{s \le \tau(\omega)\}} \mathbb{1}_{\{a_s^{\boldsymbol{\pi}}(\omega) \in \mathcal{B}_\delta(a^*)\}} \mathrm{d}s \mathbb{P}(\mathrm{d}\omega) = \int_{[t^*,T]} \mathbb{E}[\mathbb{1}_{\{s \le \tau\}} \mathbb{1}_{\{a_s^{\boldsymbol{\pi}} \in \mathcal{B}_\delta(a^*)\}}] \mathrm{d}s
$$

$$
= \int_{t^*}^T \mathbb{E}\left[\mathbb{1}_{\{s \le \tau\}} \mathbb{P}\left(a_s^{\boldsymbol{\pi}} \in \mathcal{B}_\delta(a^*) | \mathcal{F}_s\right)\right] \mathrm{d}s = \int_{t^*}^T \mathbb{E}\left[\mathbb{1}_{\{s \le \tau\}} \int_{\mathcal{B}_\delta(a^*)} \boldsymbol{\pi}(a|s, X_s^{\boldsymbol{\pi}}) \mathrm{d}a\right] \mathrm{d}s
$$

$$
\ge \min_{|x'-x^*|<\delta, |u-t^*|<\delta} \left\{ \int_{\mathcal{B}_\delta(a^*)} \boldsymbol{\pi}(a|u, x') \mathrm{d}a \right\} \int_{t^*}^T \mathbb{E}\left[\mathbb{1}_{\{s \le \tau\}}\right] \mathrm{d}s
$$

$$
= \min_{|x'-x^*|<\delta, |u-t^*|<\delta} \left\{ \int_{\mathcal{B}_\delta(a^*)} \boldsymbol{\pi}(a|u, x') \mathrm{d}a \right\} \mathbb{E}[(\tau \wedge T) - t^*] \ge 0.
$$

Since $\tau > t^*$ $\mathbb{P}$-almost surely, the above implies $\min_{|x'-x^*|<\delta, |u-t^*|<\delta} \left\{ \int_{\mathcal{B}_\delta(a^*)} \boldsymbol{\pi}(a|u, x') \mathrm{d}a \right\} = 0$. However, this contradicts Definition 1 about an admissible policy. Indeed, Definition 1-(i) stipulates $\mathrm{supp}\,\boldsymbol{\pi}(\cdot|t, x) = \mathcal{A}$ for any (t, x); hence $\int_{\mathcal{B}_\delta(a^*)} \boldsymbol{\pi}(a|t, x) \mathrm{d}a > 0$. Then the continuity in Definition 1-(ii) yields $\min_{|x'-x^*|<\delta, |u-t^*|<\delta} \left\{ \int_{\mathcal{B}_\delta(a^*)} \boldsymbol{\pi}(a|u, x') \mathrm{d}a \right\} > 0$, a contradiction. Hence we conclude $q(t, x, a; \boldsymbol{\pi}) = \hat{q}(t, x, a)$ for every (t, x, a).

(ii) Applying Itô's lemma to $e^{-\beta s} J(s, X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi})$, we get

$$
e^{-\beta s} J(s, X_s^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) - e^{-\beta t} J(t, x; \boldsymbol{\pi}) + \int_t^s e^{-\beta u}[r(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'}) - \hat{q}(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'})] \mathrm{d}u
$$

$$
= \int_t^s e^{-\beta u}\left[q(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) - \hat{q}(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'})\right] \mathrm{d}u + \int_t^s e^{-\beta u} \frac{\partial}{\partial x} J(u, X_u^{\boldsymbol{\pi}'}; \boldsymbol{\pi}) \circ \sigma(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'}) \mathrm{d}W_u.
$$

So, when $\hat{q} \equiv q(\cdot, \cdot, \cdot; \boldsymbol{\pi})$, the above process is an $(\{\mathcal{F}_s\}_{s \ge 0}, \mathbb{P})$-martingale on $[t, T]$.

(iii) Let $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$ be given satisfying the assumption in this part. It then follows from (ii) that $\int_t^s e^{-\beta u}[\hat{q}(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'}) - q(u, X_u^{\boldsymbol{\pi}'}, a_u^{\boldsymbol{\pi}'}; \boldsymbol{\pi})] \mathrm{d}u$ is an $(\{\mathcal{F}_s\}_{s \ge 0}, \mathbb{P})$-martingale. If the desired conclusion is not true, then the same argument in (i) still applies to conclude that $\min_{|x'-x^*|<\delta, |u-t^*|<\delta} \left\{ \int_{\mathcal{B}_\delta(a^*)} \boldsymbol{\pi}'(a|u, x') \mathrm{d}a \right\} = 0$, which is a contradiction because $\boldsymbol{\pi}'$ is admissible.

∎

Proof of Theorem 7

Proof

(i) We only prove the "only if" part because the "if" part is straightforward following the same argument as in the proof of Theorem 6.

So, we assume $e^{-\beta t} \hat{J}(t, X_t^{\boldsymbol{\pi}}) + \int_0^t e^{-\beta s}[r(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \hat{q}(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}})] \mathrm{d}s$ is an $(\{\mathcal{F}_s\}_{s \ge 0}, \mathbb{P})$-martingale. Hence for any initial state (t, x), we have

$$
\mathbb{E}^{\mathbb{P}}\left[e^{-\beta T} \hat{J}(T, X_T^{\boldsymbol{\pi}}) + \int_t^T e^{-\beta s}[r(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \hat{q}(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}})] \mathrm{d}s \Big| \mathcal{F}_t\right] = e^{-\beta t} \hat{J}(t, x).
$$

We integrate over the action randomization with respect to the policy $\boldsymbol{\pi}$, and then obtain

$$\mathbb{E}^{\mathbb{P}^W}\left[e^{-\beta T}\hat{J}(T,\tilde{X}_T^{\boldsymbol{\pi}}) + \int_t^T e^{-\beta s}\int_{\mathcal{A}}[r(s,\tilde{X}_s^{\boldsymbol{\pi}},a) - \hat{q}(s,\tilde{X}_s^{\boldsymbol{\pi}},a)]\boldsymbol{\pi}(a|s,\tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}s\Big|\mathcal{F}_t^W\right] = e^{-\beta t}\hat{J}(t,x)$$

This, together with the terminal condition $\hat{J}(T,x) = h(x)$, and constraint (21) yields that

$$\hat{J}(t,x) = \mathbb{E}^{\mathbb{P}^W}\left[e^{-\beta(T-t)}h(\tilde{X}_T^{\boldsymbol{\pi}}) + \int_t^T e^{-\beta(s-t)}\int_{\mathcal{A}}[r(s,\tilde{X}_s^{\boldsymbol{\pi}},a) - \gamma\log\boldsymbol{\pi}(s,\tilde{X}_s^{\boldsymbol{\pi}},a)]\boldsymbol{\pi}(a|s,\tilde{X}_s^{\boldsymbol{\pi}})\mathrm{d}a\mathrm{d}s\Big|\mathcal{F}_t^W\right].$$

Hence $\hat{J}(t,x) = J(t,x;\boldsymbol{\pi})$ for all $(t,x) \in [0,T]\times\mathbb{R}^d$. Furthermore, based on Theorem 6, the martingale condition implies that $\hat{q}(t,x,a) = q(t,x,a;\boldsymbol{\pi})$ for all $(t,x,a) \in [0,T]\times\mathbb{R}^d\times\mathcal{A}$.

(ii) This follows immediately from Theorem 6-(ii).

(iii) Let $\boldsymbol{\pi}' \in \boldsymbol{\Pi}$ be given satisfying the assumption in this part. Define

$$\hat{r}(t,x,a) := \frac{\partial\hat{J}}{\partial t}(t,x) + b(t,x,a)\circ\frac{\partial\hat{J}}{\partial x}(t,x) + \frac{1}{2}\sigma\sigma^\top(t,x,a)\circ\frac{\partial^2\hat{J}}{\partial x^2}(t,x) - \beta\hat{J}(t,x).$$

Then

$$e^{-\beta s}\hat{J}(s,X_s^{\boldsymbol{\pi}'}) - \int_t^s e^{-\beta u}\hat{r}(u,X_u^{\boldsymbol{\pi}'},a_u^{\boldsymbol{\pi}'})\mathrm{d}u$$

is an $(\{\mathcal{F}_s\}_{s\geq 0},\mathbb{P})$-local martingale, which follows from applying Itô's lemma to the above process and arguing similarly to the proof of Theorem 6-(ii). As a result, $\int_t^s e^{-\beta u}[r(u,X_u^{\boldsymbol{\pi}'},a_u^{\boldsymbol{\pi}'}) - \hat{q}(u,X_u^{\boldsymbol{\pi}'},a_u^{\boldsymbol{\pi}'}) + \hat{r}(u,X_u^{\boldsymbol{\pi}'},a_u^{\boldsymbol{\pi}'})]\mathrm{d}u$ is an $(\{\mathcal{F}_s\}_{s\geq 0},\mathbb{P})$-martingale. The same argument as in the proof of Theorem 6 applies and we conclude

$$\begin{aligned}\hat{q}(t,x,a) =& r(t,x,a) + \hat{r}(t,x,a)\\ =& \frac{\partial\hat{J}}{\partial t}(t,x) + b(t,x,a)\circ\frac{\partial\hat{J}}{\partial x}(t,x) + \frac{1}{2}\sigma\sigma^\top(t,x,a)\circ\frac{\partial^2\hat{J}}{\partial x^2}(t,x) - \beta\hat{J}(t,x) + r(t,x,a)\\ =& \frac{\partial\hat{J}}{\partial t}(t,x) + H\big(t,x,a,\frac{\partial\hat{J}}{\partial x}(t,x),\frac{\partial^2\hat{J}}{\partial x^2}(t,x)\big) - \beta\hat{J}(t,x)\end{aligned}$$

for every (t,x,a). Now the constraint (21) reads

$$\int_{\mathcal{A}}\left[\frac{\partial\hat{J}}{\partial t}(t,x) + H\big(t,x,a,\frac{\partial\hat{J}}{\partial x}(t,x),\frac{\partial^2\hat{J}}{\partial x^2}(t,x)\big) - \beta\hat{J}(t,x) - \gamma\log\boldsymbol{\pi}(a|t,x)\right]\boldsymbol{\pi}(a|t,x)\mathrm{d}a = 0,$$

for all (t,x), which, together with the terminal condition $\hat{J}(T,x) = h(x)$, is the Feynman–Kac PDE (11) for $\hat{J}$. Therefore, it follows from the uniqueness of the solution to (11) that $\hat{J} \equiv J(\cdot,\cdot;\boldsymbol{\pi})$. Moreover, it follows from Theorem 6-(iii) that $\hat{q} \equiv q(\cdot,\cdot,\cdot;\boldsymbol{\pi})$.

Finally, if $\boldsymbol{\pi}(a|t,x) = \frac{\exp\{\frac{1}{\gamma}\hat{q}(t,x,a)\}}{\int_{\mathcal{A}}\exp\{\frac{1}{\gamma}\hat{q}(t,x,a)\}\mathrm{d}a}$, then $\boldsymbol{\pi} = \mathcal{I}\boldsymbol{\pi}$ where $\mathcal{I}$ is the map defined in Theorem 2. This in turn implies $\boldsymbol{\pi}$ is the optimal policy and, hence, $\hat{J}$ is the optimal value function ∎

Proof of Proposition 8

Proof Divide both sides of (24) by γ, take exponential and then integrate over $\mathcal{A}$. Comparing the resulting equation with the exploratory HJB equation (14), we get

$$\gamma \log \left[\int_{\mathcal{A}} \exp\{\frac{1}{\gamma} q^*(t, x, a)\} \mathrm{d}a \right] = 0.$$

This yields (25). The expression (26) follows then from (13). ∎

Proof of Theorem 9

Proof

(i) Let $\widehat{J^*} = J^*$ and $\widehat{q^*} = q^*$ be the optimal value function and optimal q-function respectively. For any $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, applying Itô's lemma to the process $e^{-\beta s} J^*(s, X_s^{\boldsymbol{\pi}})$, we obtain for $0 \le t < s \le T$:

$$e^{-\beta s} J^*(s, X_s^{\boldsymbol{\pi}}) - e^{-\beta t} J^*(t, x) + \int_t^s e^{-\beta u} [r(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}) - q^*(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}})] \mathrm{d}u$$

$$= \int_t^s e^{-\beta u} \left[\frac{\partial J^*}{\partial t}(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}) + H\left(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}, \frac{\partial J^*}{\partial x}(u, X_u^{\boldsymbol{\pi}}), \frac{\partial^2 J^*}{\partial x^2}(u, X_u^{\boldsymbol{\pi}}) - \beta J^*(u, X_u^{\boldsymbol{\pi}}) \right. $$

$$\left. - q^*(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}) \right] \mathrm{d}u + \int_t^s e^{-\beta u} \frac{\partial}{\partial x} J^*(u, X_u^{\boldsymbol{\pi}}) \circ \mathrm{d}W_u$$

$$= \int_t^s e^{-\beta u} \frac{\partial}{\partial x} J^*(u, X_u^{\boldsymbol{\pi}}) \circ \sigma(u, X_u^{\boldsymbol{\pi}}, a_u^{\boldsymbol{\pi}}) \mathrm{d}W_u,$$

where the $\int \cdots \mathrm{d}u$ term vanishes due to the definition of q^* in (24). Hence (28) is a martingale.

(ii) The second constraint in (27) implies that $\widehat{\boldsymbol{\pi}^*}(a|t, x) := \exp\{\frac{1}{\gamma} \widehat{q^*}(t, x, a)\}$ is a probability density function, and $\widehat{q^*}(t, x, a) = \gamma \log \widehat{\boldsymbol{\pi}^*}(a|t, x)$. So $\widehat{q^*}(t, x, a)$ satisfies the second constraint in (21) with respect to the policy $\widehat{\boldsymbol{\pi}^*}$. When (28) is an $(\{\mathcal{F}_s\}_{s \ge 0}, \mathbb{P})$-martingale under the given admissible policy $\boldsymbol{\pi}$, it follows from Theorem 7–(iii) that $\widehat{J^*}$ and $\widehat{q^*}$ are respectively the value function and the q-function associated with $\widehat{\boldsymbol{\pi}^*}$. Then the improved policy is $\mathcal{I}\widehat{\boldsymbol{\pi}^*}(a|t, x) := \frac{\exp\{\frac{1}{\gamma}\widehat{q^*}(t,x,a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}\widehat{q^*}(t,x,a)\}\mathrm{d}a} = \exp\{\frac{1}{\gamma}\widehat{q^*}(t, x, a)\} = \widehat{\boldsymbol{\pi}^*}(a|t, x)$. However, Theorem 2 yields that $\widehat{\boldsymbol{\pi}^*}$ is optimal, completing the proof. ∎

Proof of Theorem 10

Proof Let us compute the KL-divergence:

$$\int_{\mathcal{A}} \big[\log \boldsymbol{\pi}'(a|t,x) - \frac{1}{\gamma} H\big(t,x,a, \frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\big] \boldsymbol{\pi}'(a|t,x)\mathrm{d}a$$

$$=D_{KL}\left(\boldsymbol{\pi}'(\cdot|t,x)\Big|\Big| \exp\{\frac{1}{\gamma}H\big(t,x,\cdot, \frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi}))\}\right)$$

$$\leq D_{KL}\left(\boldsymbol{\pi}(\cdot|t,x)\Big|\Big| \exp\{\frac{1}{\gamma}H\big(t,x,\cdot, \frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi}))\}\right)$$

$$= \int_{\mathcal{A}} \big[\log \boldsymbol{\pi}(a|t,x) - \frac{1}{\gamma} H\big(t,x,a, \frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\big] \boldsymbol{\pi}(a|t,x)\mathrm{d}a.$$

Hence $\int_{\mathcal{A}} \big[q(t,x,a;\boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}'(a|t,x)\big]\boldsymbol{\pi}'(a|t,x)\mathrm{d}a \geq \int_{\mathcal{A}} \big[q(t,x,a;\boldsymbol{\pi}) - \gamma \log \boldsymbol{\pi}(a|t,x)\big]\boldsymbol{\pi}(a|t,x)\mathrm{d}a$. Following the same argument as in the proof of Theorem 2, we conclude $J(t,x;\boldsymbol{\pi}') \geq J(t,x;\boldsymbol{\pi})$. ∎

Proof of Lemma 11

Proof The first statement has been prove in Jia and Zhou (2022b, Lemma 7). We focus on the proof of the second statement, which is similar to that of Theorem 2.

For the two admissible policies $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$, we apply Itô's lemma to the value function under $\boldsymbol{\pi}$ along the process $\tilde{X}^{\boldsymbol{\pi}'}$ to obtain

$$J(\tilde{X}_u^{\boldsymbol{\pi}'};\boldsymbol{\pi}) - J(\tilde{X}_t^{\boldsymbol{\pi}'};\boldsymbol{\pi}) + \int_t^u \int_{\mathcal{A}} [r(\tilde{X}_s^{\boldsymbol{\pi}'},a) - \gamma \log \boldsymbol{\pi}'(a|\tilde{X}_s^{\boldsymbol{\pi}'})]\boldsymbol{\pi}'(a|\tilde{X}_s^{\boldsymbol{\pi}'})\mathrm{d}a\mathrm{d}s$$

$$= \int_t^u \int_{\mathcal{A}} \big[H\big(\tilde{X}_s^{\boldsymbol{\pi}'},a, \frac{\partial J}{\partial x}(\tilde{X}_s^{\boldsymbol{\pi}'};\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(\tilde{X}_s^{\boldsymbol{\pi}'};\boldsymbol{\pi})\big) - \gamma \log \boldsymbol{\pi}'(a|X_s^{\boldsymbol{\pi}'})\big]\boldsymbol{\pi}'(a|\tilde{X}_s^{\boldsymbol{\pi}'})\mathrm{d}a\mathrm{d}s$$

$$+ \int_t^u \frac{\partial}{\partial x} J(\tilde{X}_s^{\boldsymbol{\pi}'};\boldsymbol{\pi}) \circ \tilde{\sigma}\big(\tilde{X}_s^{\boldsymbol{\pi}'},\boldsymbol{\pi}'(\cdot|\tilde{X}_s^{\boldsymbol{\pi}'})\big)\mathrm{d}W_s$$

$$\geq \int_t^u \int_{\mathcal{A}} \big[H\big(\tilde{X}_s^{\boldsymbol{\pi}'},a, \frac{\partial J}{\partial x}(\tilde{X}_s^{\boldsymbol{\pi}'};\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(\tilde{X}_s^{\boldsymbol{\pi}'};\boldsymbol{\pi})\big) - \gamma \log \boldsymbol{\pi}(a|X_s^{\boldsymbol{\pi}'})\big]\boldsymbol{\pi}(a|\tilde{X}_s^{\boldsymbol{\pi}'})\mathrm{d}a\mathrm{d}s$$

$$+ \int_t^u \frac{\partial}{\partial x} J(\tilde{X}_s^{\boldsymbol{\pi}'};\boldsymbol{\pi}) \circ \tilde{\sigma}\big(\tilde{X}_s^{\boldsymbol{\pi}'},\boldsymbol{\pi}'(\cdot|\tilde{X}_s^{\boldsymbol{\pi}'})\big)\mathrm{d}W_s$$

$$=V(\boldsymbol{\pi})(u-t) + \int_t^u \frac{\partial}{\partial x} J(\tilde{X}_s^{\boldsymbol{\pi}'};\boldsymbol{\pi}) \circ \tilde{\sigma}\big(\tilde{X}_s^{\boldsymbol{\pi}'},\boldsymbol{\pi}'(\cdot|\tilde{X}_s^{\boldsymbol{\pi}'})\big)\mathrm{d}W_s,$$

where the inequality is due to the same argument as in the proof of Theorem 10, and the last equality is due to the Feynman–Kac formula (38).

Therefore, after a localization argument, we have

$$\frac{1}{T}\mathbb{E}\left[J(\tilde{X}_T^{\boldsymbol{\pi}'};\boldsymbol{\pi}) - J(x;\boldsymbol{\pi}) + \int_0^T \int_{\mathcal{A}} [r(\tilde{X}_s^{\boldsymbol{\pi}'},a) - \gamma \log \boldsymbol{\pi}'(a|\tilde{X}_s^{\boldsymbol{\pi}'})]\boldsymbol{\pi}'(a|\tilde{X}_s^{\boldsymbol{\pi}'})\mathrm{d}a\mathrm{d}s\Big|\tilde{X}_0^{\boldsymbol{\pi}'}=x\right] \geq V(\boldsymbol{\pi}).$$

Taking the limit $T \to \infty$, we obtain $V(\boldsymbol{\pi}') \geq V(\boldsymbol{\pi})$. ∎

Proof of Theorem 12

Proof The proof is parallel to that of Theorems 6 and 7, which is omitted. ∎

References

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

L. C. Baird. Reinforcement learning in continuous time: Advantage updating. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 4, pages 2448–2453. IEEE, 1994.

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.

G. Brunick and S. Shreve. Mimicking an Itô process by a solution of a stochastic differential equation. *The Annals of Applied Probability*, 23(4):1584–1628, 2013.

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.

J. Fan, Z. Wang, Y. Xie, and Z. Yang. A theoretical analysis of deep Q-learning. In *Learning for Dynamics and Control*, pages 486–489. PMLR, 2020.

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

X. Gao, Z. Q. Xu, and X. Y. Zhou. State-dependent temperature control for Langevin diffusions. *SIAM Journal on Control and Optimization*, pages 1–26, 2020.

S. Gu, T. Lillicrap, I. Sutskever, and S. Levine. Continuous deep Q-learning with model-based acceleration. In *International Conference on Machine Learning*, pages 2829–2838. PMLR, 2016.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018a.

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*, 2018b.

X. Han, R. Wang, and X. Y. Zhou. Choquet regularization for reinforcement learning. *arXiv preprint arXiv:2208.08497*, 2022.

Y. Jia and X. Y. Zhou. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *Journal of Machine Learning Research*, 23(198): 1–55, 2022a.

Y. Jia and X. Y. Zhou. Policy gradient and actor-critic learning in continuous time and space: Theory and algorithms. *Journal of Machine Learning Research*, 23(275):1–50, 2022b.

I. Karatzas and S. Shreve. *Brownian Motion and Stochastic Calculus*, volume 113. Springer, 2014.

J. Kim, J. Shin, and I. Yang. Hamilton-Jacobi deep Q-learning for deterministic continuous-time systems with Lipschitz continuous controls. *Journal of Machine Learning Research*, 22:206–1, 2021.

A. E. Lim and X. Y. Zhou. Mean-variance portfolio selection with random parameters in a complete market. *Mathematics of Operations Research*, 27(1):101–120, 2002.

X. Mao. *Stochastic Differential Equations and Applications*. Woodhead, 2 edition, 2007. ISBN 978-1-904275-34-3.

F. S. Melo, S. P. Meyn, and M. I. Ribeiro. An analysis of reinforcement learning with function approximation. In *Proceedings of the 25th International Conference on Machine Learning*, pages 664–671, 2008.

S. P. Meyn and R. L. Tweedie. Stability of Markovian processes III: Foster–Lyapunov criteria for continuous-time processes. *Advances in Applied Probability*, 25(3):518–548, 1993.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

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.

B. O'Donoghue. Variational Bayesian reinforcement learning with regret bounds. *Advances in Neural Information Processing Systems*, 34:28208–28221, 2021.

I. Osband, B. Van Roy, D. J. Russo, and Z. Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

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

D. Shah and Q. Xie. Q-learning with nearest neighbors. *Advances in Neural Information Processing Systems*, 31, 2018.

Y. Sun. The exact law of large numbers via Fubini extension and characterization of insurable risks. *Journal of Economic Theory*, 126(1):31–69, 2006.

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

C. Tallec, L. Blier, and Y. Ollivier. Making deep Q-learning methods robust to time discretization. In *International Conference on Machine Learning*, pages 6096–6104. PMLR, 2019.

W. Tang, Y. P. Zhang, and X. Y. Zhou. Exploratory HJB equations and their convergence. *SIAM Journal on Control and Optimization*, 60(6):3191–3216, 2022.

M. Uehara, C. Shi, and N. Kallus. A review of off-policy evaluation in reinforcement learning. *arXiv preprint arXiv:2212.06355*, 2022.

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.

C. J. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3):279–292, 1992.

C. J. C. H. Watkins. Learning from delayed rewards. *PhD thesis, Cambridge University*, 1989.

W. M. Wonham. On the separation theorem of stochastic control. *SIAM Journal on Control*, 6(2):312–326, 1968.

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

X. Y. Zhou. Curse of optimality, and how do we break it. *SSRN preprint SSRN 3845462*, 2021.

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.

X. Y. Zhou and G. Yin. Markowitz's mean-variance portfolio selection with regime switching: A continuous-time model. *SIAM Journal on Control and Optimization*, 42(4):1466–1482, 2003.

S. Zou, T. Xu, and Y. Liang. Finite-sample analysis for SARSA with linear function approximation. *Advances in Neural Information Processing Systems*, 32, 2019.