PhiBE-Q-Learning: Bridging Off-Policy Reinforcement Learning and Continuous-Time Control

Yutong Ren * Yuhua Zhu †

Abstract

In this paper, we develop an off-policy method for continuous-time reinforcement learning (CTRL), where the system dynamics are governed by an unknown stochastic differential equation (SDE) and only discrete-time trajectory data are available. A central challenge is that the classical state–action value function $Q(s, a)$, which enables off-policy learning in discrete-time RL, does not exist in CTRL (Baird, 1994, Jia and Zhou, 2023, Tallec et al., 2019). On the other hand, continuous-time control provides local notions such as the instantaneous advantage function $q(s, a)$, but these typically rely on on-policy data. To address this, we introduce a new definition of the state–action value function in CTRL and derive its governing equation.

Building on the PhiBE approximation (Zhu, 2024, Zhu et al., 2025), we propose iterative algorithms to approximate the optimal Q-function in both model-based and model-free settings using only discrete-time off-policy data. Under linear function approximation, we establish convergence guarantees and derive explicit convergence rates for the proposed method.

1 Introduction

Reinforcement learning (RL) has achieved remarkable success in the digital world, including AlphaGo (Silver et al., 2016), Atari gameplay (Mnih et al., 2015), and the fine-tuning of large language models (Ziegler et al., 2019). A key feature underlying these successes is that the system state remains static between observations. This property is natural in digital settings, making such environments well suited to RL algorithms built on the Markov decision process (MDP) framework. However, decision-making in the physical world departs from this digital paradigm in two fundamental respects. First, the objective is not merely to maximize rewards at the discrete times when data are observed, but to control performance continuously over time, including during periods without observations. In dynamic treatment regimes, for example, a patient's physiological state (e.g., glucose level) evolves continuously, while measurements are only taken intermittently. The clinical goal, however, is to regulate

*[1]Department of Statistics and Data Science, University of California Los Angeles, California, USA. ytren@ucla.edu

†[2]Department of Statistics and Data Science, University of California, Los Angeles, USA. yuhuazhu@ucla.edu

the state at all times (Battelino et al., 2019, Cobelli et al., 2009, Emerson et al., 2023). RL in the physical world thus poses a substantially more challenging objective than its classical counterpart. Second, many physical systems exhibit structured and smooth continuous-time dynamics, a feature that is largely absent in digital environments and not exploited by the MDP framework. RL in the physical world are also known as continuous-time reinforcement learning (CTRL) (Wang et al., 2020).

Off-policy learning is equally critical in real-world applications (Ma et al., 2020, Sutton et al., 1998, Watkins and Dayan, 1992). Off-policy data consists of trajectories generated under a behavior policy that differs from the target policy, and the behavior policy may be unknown. Leveraging off-policy data can substantially improve sample efficiency, particularly in domains where data collection is costly, such as healthcare. Moreover, it enables the use of existing datasets, which is critical in settings where exploration is unsafe, such as autonomous driving.

The central question this paper studies is: what is a principled framework for CTRL when the system evolves according to an unknown stochastic differential equation (SDE), yet only discrete-time off-policy observations are available?

Existing work has largely followed two directions. The first approach seeks to reconstruct the underlying continuous-time dynamics from discrete observations and subsequently solve the induced optimal control problem using tools from stochastic control (Agarwal et al., 2020, Kamalapurkar et al., 2016, Yang et al., 2014, Yildiz et al., 2021). This approach has two appealing features. First, the system dynamics depend only on the state and action, therefore, it is independent of the policy, making it naturally suited for leveraging off-policy data. Second, it preserves the continuous-time structure of the problem. However, this approach also faces two fundamental limitations. First, identifying a continuous-time model from discrete observations is inherently ill-posed: in general, infinitely many continuous-time dynamics can induce the same discrete-time transition law (Zhu et al., 2025). As a result, model misspecification is unavoidable in practice and may lead to suboptimal policies. Second, the approach decomposes the problem into model identification and control optimization, each introducing its own source of error. When the ultimate goal is to find the optimal policy, explicitly estimating the underlying dynamics may be unnecessary and potentially inefficient.

The second approach instead reformulates CTRL as a discrete-time MDP and applies standard model-free off-policy RL algorithms, such as Q-learning or conversative Q-learning (Kumar et al., 2020, Watkins and Dayan, 1992). These formulations rely solely on discrete-time transition dynamics, thereby avoiding the unidentifiability issues inherent in the first approach. In addition, model-free RL methods directly learn the optimal policy without explicitly estimating the underlying dynamics, providing a simple and broadly applicable framework across different systems. However, this approach also has two notable limitations. First, discrete-time RL algorithms do not exploit the SDE structure of the underlying dynamics. Instead, they treat all systems the same, leading to inefficient use of information and potentially suboptimal policies. More specifically, the learning error can be highly sensitive to both system parameters and reward functions (Zhu et al., 2025). Second, off-policy learning in the MDP framework is typically based on the action–value function $Q(s, a)$ (Watkins and Dayan, 1992). As observed in (Baird, 1994, Jia and Zhou, 2023, Tallec et al., 2019), in the continuous-time setting, the state-action $Q(s, a)$ function can be viewed as a small perturbation around the state value function $V(s)$, implying that its variation across

actions is small, which makes it difficult to accurately distinguish between actions. To address this issue, these papers propose a rescaled advantage function $q(s, a)$ to amplify variation in the action space. However, this construction depends on V and thus cannot be directly applied in the off-policy setting (see Section 3 for a detailed discussion).

We tackle this challenge in two steps. First, we introduce a new definition of the state–action value function Q. In classical RL, a key advantage of the Q-function is its ability to handle off-policy data, whereas the value function V does not share this property. This distinction arises because the discrete-time Bellman equation for Q depends only on the transition distribution under a given action a, while the Bellman equation for V depends on the transition induced by a policy π. Our first contribution is to define a continuous-time Q-function that preserves this advantage. Specifically, the proposed Q-function satisfies two properties: (i) it does not degenerate to the state value function V, unlike standard continuous-time limits of RL formulations; and (ii) its governing equation depends only on the dynamics under action a, not on any policy π.

The second step is to develop a new iterative algorithm to approximate the optimal value function Q^*. We begin with an algorithm for the setting where the underlying dynamics are known, and provide a rigorous error analysis with convergence rates under linear function approximation. Building on this, and leveraging the PhiBE framework (Zhu, 2024, Zhu et al., 2025), we propose a continuous-time analogue of model-free Q-learning (Watkins and Dayan, 1992). This is the second key contribution of this paper. The proposed algorithm for off-policy continuous-time RL combines the strengths of continuous-time control and classical RL while avoiding their key limitations. Compared to the model-based control approaches, it relies only on discrete-time transition data and does not require explicit estimation of the underlying dynamics. In addition, it is a model-free algorithm in the sense that the update rule is invariant across different systems. Compared to standard RL algorithms, it explicitly exploits the SDE structure of the dynamics, integrating discrete-time observations with continuous-time evolution. Moreover, the resulting Q-function is not a vanishing perturbation of the value function V, which makes the formulation well-posed and numerically stable in the continuous-time limit.

Contributions. We summarize the main contributions as follows.

- We introduce a new state–action value function Q for CTRL and derive its governing equation, which enables off-policy learning in continuous time.
- We develop an iterative algorithm to approximate the optimal state–action value function under known dynamics, and establish error bounds and convergence rates under linear function approximation that remain well-posed as the time discretization tends to 0.
- When the dynamics are unknown, we propose *PhiBE-Q-Learning*, a model-free algorithm that effectively leverages off-policy data and the SDE structure for CTRL problems.

1.1 Related work

1.1.1 standard Discrete-time RL

Classical reinforcement learning (RL) methods such as Q-learning (Watkins and Dayan, 1992), DQN (Mnih et al., 2015), provide a powerful framework for off-policy learning in discrete-time setting. Our approach draws inspiration from these algorithms. Our update rule is an analogue of Q-learning (Watkins and Dayan, 1992) in continuous-time.

In discrete-time Q-learning, convergence typically relies on the discount factor $\gamma < 1$, which ensures that the optimal Bellman operator is a contraction mapping (Bertsekas, 2012). In the continuous-time setting, the corresponding discount factor takes the form $\gamma = e^{-\beta \Delta t} \to 1$ as the time discretization $\Delta t \to 0$. This limit would, in principle, destroy the contraction property and lead to ill-conditioned or diverging guarantees. In this paper, we refine the error bounds in terms of Δt. In particular, we establish convergence and stability results that are uniform in Δt. We show that, when the underlying dynamics follow an SDE, the additional regularity of the system yields well-conditioned guarantees even $\Delta t \to 0$. Similar results are also derived in other CTRL papers under different settings (Mou and Zhu, 2024, Zhu, 2024, Zhu et al., 2025).

1.1.2 Continuous-time RL(CTRL)

A number of recent works have investigated continuous-time reinforcement learning (CTRL), where the system dynamics are modeled by SDEs (Jia and Zhou, 2022a,b, Kim and Yang, 2020, Szpruch et al., 2024, Wang and Zhou, 2020). In particular, (Wang et al., 2020) connects CTRL with stochastic control with several extensions developed in subsequent works (Jia and Zhou, 2022b, Szpruch et al., 2024). These methods typically enjoy strong theoretical guarantees and exploit the SDE structure when continuous-time data are available. However, they face three major limitations in settings with discrete-time off-policy data. First, when applied to discrete-time observations, these algorithms do not fully exploit the SDE structure of the underlying dynamics. Second, many approaches depend on long trajectories spanning from an initial state to a terminal state, which restricts applicability when only fragmented or short trajectories are available. Finally, the q function defined in (Baird, 1994, Jia and Zhou, 2023, Tallec et al., 2019) remains dependent on the value function V, preventing straightforward use of off-policy data. These limitations motivate the development of methods that can learn directly from discrete observations, leverage off-policy samples in CTRL.

1.1.3 PhiBE

To enable learning directly from discrete trajectory data, (Zhu, 2024) proposed the PhiBE framework, which explicitly accounts for discretization effects and leverages the SDE structure. By doing so, PhiBE improves the accuracy and efficiency of learning from discrete trajectory data in stochastic continuous-time systems. At the same time, PhiBE still leaves certain aspects unaddressed. In particular, existing PhiBE algorithms do not fully support learning from off-policy data. This work extends the PhiBE framework to address the off-policy data utilization. Our method aims to find the solution of PhiBE instead of the original HJB

equation, thereby retaining PhiBE's key advantage of handling the discrete trajectory data. Moreover, by introducing a new definition of the continuous-time action-value function Q, we derive an analogue of the optimal Bellman equation on Q independent of policy π. This formulation allows the algorithm to utilize trajectories generated under different policies.

Finally, we emphasize that our attention in this paper is restricted to the convergence rate when there are enough data. The finite sample error (Muehlebach et al., 2025, Zhao et al., 2025, Zhu, 2024) is left for future work.

1.1.4 Classical HJB solvers

Our iterative method under known dynamics shares the same intuition as classical numerical schemes for HJB equation (Feng et al., 2013, Schaeffer and Hou, 2016). However, the key difference lies in the formulation and convergence mechanism. Traditional finite-difference methods approximate the HJB equation by discretizing the state space on a grid, which requires careful mesh design and often imposes CFL-type constraints on the grid spacing to ensure stability and convergence. In contrast, our approach avoids explicit spatial discretization by approximating the Galerkin projection directly from trajectory data. This eliminates the need for grid-based computation and CFL conditions, making the method more flexible and better suited for data-driven implementations in CTRL.

Moreover, based on the PhiBE framework, our method could extend to a model-free algorithm when the dynamics are unknown. Specifically, it can directly approximate the action-value function without requiring explicit knowledge of the system dynamics. Here "model-free" means that the update rule remains unchanged across different forms of underlying dynamics, allowing the algorithm to adapt automatically without model-specific modifications.

1.2 Notation and Organization

Notation We define weighted L^2-norm as follows

$$\langle f, g\rangle = \int f(s,a)g(s,a)\mu(s,a)dsda, \quad \|f\| = \langle f, f\rangle .$$

Here, $\mu(s,a)$ denotes the probability density of the off-policy data distribution, supported on a compact set $\Omega \subset \mathbb{R}^d \times \mathbb{R}^m$.

For vector $x \in \mathbb{R}^n$, and function $f(s,a)$, we define $\|\cdot\|_2, \|\cdot\|_\infty$ as follows,

$$\|x\|_2^2 = \sum_{i=1}^n x_i^2, \quad \|f\|_\infty = \sup_{(s,a)\in\Omega} \|f\|_2$$

If no specific remark are given, then ∇, ∇^2 refer to ∇_s, ∇_s^2.

Organization. Section 2 introduces the problem setup. Section 3 presents the new state–action Q-function. Section 3.2 develops the algorithm under known dynamics and establishes its theoretical guarantees, while Section 3.3 derives the model-free algorithm for unknown dynamics. Section 4 reports numerical experiments that validate the theory.

2 Preliminary

2.1 Stochastic optimal control

We consider a continuous-time stochastic optimal control problem defined on a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, \mathbb{P})$. The state process $\{s_t\}_{t\geq 0}$ takes values in $\mathbb{R}^d$ and evolves according to a controlled stochastic differential equation (SDE):

$$ds_t = b(s_t, a_t)\, dt + \sigma(s_t, a_t)\, dB_t, \tag{1}$$

where $b : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d$ is the drift term, $\sigma : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^{d\times k}$ is the diffusion term, and B_t is a k-dimensional standard Brownian motion. At time t, the agent takes an action $a_t = \pi(s_t)$ according to a feedback control. Here the feedback control is defined as a deterministic mapping $\pi : \mathbb{R}^d \to \mathcal{A}$ from the state space to a compact action space $\mathcal{A} \subseteq \mathbb{R}^m$. Given a fixed policy π and initial state $s_0 = s$, the value function is defined as

$$V^\pi(s) = \mathbb{E}\left[\int_0^\infty e^{-\beta t} r(s_t, \pi(s_t))\, dt \;\middle|\; s_0 = s\right], \tag{2}$$

where $\beta > 0$ is the discount rate. The goal of stochastic optimal control problem is to find the optimal feedback control that maximizes the value function,

$$\pi^*(s) = \text{argmax}_\pi\, V^\pi(s), \quad V^*(s) = V^{\pi^*}(s). \tag{3}$$

We assume the following assumption to ensure the well-posedness of the above stochastic control problem.

Assumption 1. *(i) The action space $\mathcal{A}$ is compact; the functions $b(s,a)$, $\sigma(s,a)$, and $r(s,a)$ are continuous in a, and locally uniformly Lipschitz continuous in s. The reward function $r(s,a)$ is uniformly bounded.*

Moreover, the optimal value function $V^*(s)$ and the optimal feedback policy π^* defined in (3) satisfy

$$\beta V^*(s) = \sup_{a\in\mathcal{A}}[r(s,a) + (\mathcal{L}_{b,\Sigma}V^*)(s,a)], \quad \pi^*(s) = \arg\sup_{a\in\mathcal{A}}[r(s,a) + (\mathcal{L}_{b,\Sigma}V^*)(s,a)],$$

Here

$$\mathcal{L}_{b,\Sigma} = b(s,a)\cdot\nabla_s + \frac{1}{2}\Sigma(s,a) : \nabla_s^2 \tag{4}$$

with $\Sigma(s,a) = \sigma(s,a)\sigma^\top(s,a) \in \mathbb{R}^{d\times d}$ and $\Sigma(s,a) : \nabla_s^2 = \Sigma_{i,j}\Sigma(s,a)_{ij}\partial_{s_i}\partial_{s_j}$. We refer readers to (Fleming and Soner, 2006, Pham, 2009, Yong and Zhou, 1999) for detailed accounts of the classical stochastic control theory.

2.2 Reinforcement learning

In the setting of reinforcement learning, the dynamics of the environment, i.e., $b(s,a), \sigma(s,a)$ are unknown. We assume access to discretized information, either in the form of transition

densities

$$\rho_{\Delta t}(s' \mid s, a), \qquad \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{A} \to [0, \infty),$$

which denotes the probability density function of $s_{\Delta t}$ given $s_0 = s$ and $a(\tau) = a$ for $\tau \in (0, \Delta t)$, or a dataset of transitions

$$\{(s_i, a_i, s_i', r_i)\}_{i=1}^n, \qquad (s_i, a_i) \sim \mu(s, a), \quad s_i' \sim \rho_{\Delta t}(\cdot \mid s_i, a_i), \quad r_i = r(s_i, a_i). \tag{5}$$

Here the actions $\{a_i\}_{i=1}^n$ are generated by some (possibly unknown) behavior policy and are not controlled by the learner. Note that we consider a piecewise constant control where the action is frozen at the left endpoint, i.e., $a_t = a_0$ for $t \in [0, \Delta t]$ in the data set, while the goal is still to find the optimal feedback control $\pi^*(s)$ defined in (3).

2.3 PhiBE

As indicated in (Zhu, 2024, Zhu et al., 2025), the optimal-PhiBE provides a better approximation to the continuous-time optimal control problem compared to the Bellman equation when the underlying dynamics are smooth or the reward function oscillates, as it preserves the underlying SDE structure. Therefore, in this work, we adopt PhiBE to approximate the value function. The first-order Optimal-PhiBE corresponding to (Zhu et al., 2025) is given by

Definition 1 (PhiBE)**.**

$$\beta \hat{V}^*(s) = \sup_{a \in \mathcal{A}} \left\{ r(s, a) + \mathcal{L}_{\hat{b}, \hat{\Sigma}} \hat{V}^*(s) \right\}, \tag{6}$$

where

$$\begin{aligned} \hat{b}(s, a) &= \frac{1}{\Delta t} \int (s' - s) \rho_{\Delta t}(s' | s, a) ds' \\ \hat{\Sigma}(s, a) &= \frac{1}{\Delta t} \int (s' - s)(s' - s)^\top \rho_{\Delta t}(s' | s, a) ds' \end{aligned} \tag{7}$$

As shown in Theorem 3.5 of (Zhu et al., 2025), under suitable regularity conditions on the dynamics b and σ, the optimal policy $\hat{\pi}^*(s)$ derived from PhiBE achieves an $O(\Delta t)$ approximation to the true optimal policy, when evaluated in terms of the true value function under the respective policies. We will use the PhiBE formulation in Section 3.3 when only discrete-time information are available.

3 PhiBE Q-Learning

In this section, we aim to find a method leverages the off-policy data. Off-policy data refers to experience collected under a behavior policy that is different from the target policy being optimized, in which the behavior policy may be unknown. Off-policy learning plays a crucial role in continuous-time reinforcement learning (CTRL). By allowing the use of data generated under arbitrary behavior policies, it significantly improves sample efficiency and makes it possible to reuse trajectories, which is especially important when collecting new

data is costly. Furthermore, in many practical applications such as healthcare or autonomous driving, exploration is inherently risky. Off-policy learning provides a way to leverage existing datasets without requiring unsafe interactions. It also helps reduce the limitation caused by using data only from the current policy, enabling learning to reach an optimal policy.Finally, the ability to separate safe data collection from optimal policy optimization highlights the practical necessity of off-policy methods for CTRL, motivating our extension of PhiBE to this setting.

In standard RL, the key reason why $Q(s,a)$ naturally supports off-policy learning is because the Bellman equation

$$Q^\pi(s,a) = r(s,a) + \gamma\, \mathbb{E}_{s'\sim\rho(\cdot|s,a)} \left(\mathbb{E}_{a'\sim\pi(\cdot|s)} \left[Q^\pi(s',a') \right] \right) \tag{8}$$

is determined by the unknown transition dynamics $\rho(\cdot|s,a)$ independent of the target policy π. While the Bellman equation for the value functions $V^\pi(s)$

$$V^\pi(s) = r^\pi(s) + \gamma\, \mathbb{E}_{s'\sim\rho^\pi(\cdot|s)} \left(V^\pi(s') \right)$$

is determined by the unknown marginal transition kernel $\rho^\pi(s'|s) = \int \pi(a|s)\, \rho(s'|s,a)\, da$, which explicitly depends on π. As a result, when data are generated from a different behavior policy, the estimation of V^π is inherently biased if the off-policy data is directly applied. In contrast, the Bellman optimality equation for $Q^*(s,a)$ is conditioned directly on (s,a), and the transition kernel $\rho(s'|s,a)$ depends only on the environment dynamics, not on π. Thus, any sample (s,a,s',r) collected under an arbitrary behavior policy provides an unbiased estimate of the right-hand side of the Bellman equation, making off-policy learning feasible for Q.

Although the Bellman equation for $V^\pi(s)$ can, in principle, be adapted to off-policy learning by applying importance sampling to correct for the mismatch between the target policy π and the behavior policy, this approach faces two challenges in practice. First, importance sampling often leads to high variance, especially in long-horizon problems or when the behavior and target policies differ substantially. Second, the behavior policy itself may be unknown, making it difficult to compute the required importance sampling ratios. These limitations motivate the use of Q-functions, whose Bellman equation is naturally off-policy without requiring such corrections.

A direct generalization of the discrete-time action-value function $Q^\pi(s,a)$ to the continuous-time setting is not feasible. As proved in (Jia and Zhou, 2023, Tallec et al., 2019), the discrete-time action-value function $Q^\pi_{\Delta t}(s,a)$ defined as

$$Q^\pi_{\Delta t}(s,a) = \mathbb{E}_{\substack{a_i\sim\pi(\cdot|s_i),i\geq 1 \\ s_{i+1}\sim\rho_{\Delta t}(\cdot|s_i,a_i),i\geq 0}} \Big[\sum_{i=0}^{\infty} \gamma^i r(s_i,a_i) | s_0 = s, a_0 = a \Big] \tag{9}$$

will converge to V^π as $\Delta t \to 0$. The reason is that, as the discretization step $\Delta t \to 0$, the duration over which an action influences the environment also vanishes. In this limit, the effect of choosing action a at state s becomes negligible, and the action-value function $Q^\pi(s,a)$ effectively degenerates to the value function $V^\pi(s)$.

In (Jia and Zhou, 2023, Tallec et al., 2019), the authors propose an action-value function

defined as

$$q^\pi(s,a) := r(s,a) + [\mathcal{L}_{b,\Sigma} V^\pi](s,a) - \beta V^\pi(s),$$

where $[\mathcal{L}_{b,\Sigma} f](s,a) := b(s,a) \cdot \nabla f(s) + \frac{1}{2}\Sigma(s,a) : \nabla^2 f(s)$. However, this formulation depends on V^π, which still requires on-policy transitions generated by π. Therefore, to handle the off-policy data setting, we introduce a modified definition of the action-value function in the CTRL framework.

In Sections 3.1 and 3.2, we begin by analyzing the known-dynamics setting, where the continuous transition dynamics (b, Σ) are assumed to be known. Moreover, we derive an iterative algorithm and establish the exponential convergence of the proposed scheme. Section 3.3.1 further demonstrates that PhiBE allows us to extend the analysis from known dynamics to the unknown-dynamics case, where only discrete transition dynamics $\rho_{\Delta t}(s'|s,a)$ are available. Finally, Section 3.3.2 also presents a model-free, off-policy data-driven algorithm that implements the proposed method when the dynamics are unknown and only discrete-time data is available, followed by numerical experiments in Section 4 to demonstrate the performance of the approach.

3.1 Continuous-time state-action value function $Q(s,a)$

Definition 2 (Continuous-time Bellman equation for $Q^\pi(s,a)$)**.**

$$Q^\pi(s,a) = r(s,a) + \mathcal{L}_{b,\Sigma}\left[\mathbb{E}_{a\sim\pi}(Q^\pi)\right](s,a) + (1-\beta)\mathbb{E}_{a\sim\pi}[Q^\pi(s,a)] \tag{10}$$

Several remarks on the definition of Q^π are in order.

First, comparing (10) with the continuous-time Bellman equation for V^π defined in (2),

$$\beta V^\pi(s) = r^\pi(s) + \mathcal{L}_{b^\pi,\Sigma^\pi} V^\pi(s), \quad b^\pi(s) = \mathbb{E}_{a\sim\pi}[b(s,a)], \quad \Sigma^\pi(s) = \mathbb{E}_{a\sim\pi}[\Sigma(s,a)],$$

we see a key structural difference. The operator in (10) is governed by the dynamics $(b(s,a), \Sigma(s,a))$, and therefore does not depend on the target policy π. This mirrors the discrete-time Bellman equation for $Q_{\Delta t}$ in (8). In contrast, the dynamics for V^π depend on (b^π, Σ^π) and thus on π. Consequently, when the dynamics are unknown, Q^π naturally accommodates off-policy data, whereas V^π does not.

Second, the continuous-time Q^π can also be viewed as the sum of the value function and the instantaneous advantage function:

$$Q^\pi(s,a) = \lim_{\Delta t\to 0}\left[\frac{Q^\pi_{\Delta t}(s,a) - V^\pi_{\Delta t}(s)}{\Delta t}\right] + V^\pi(s),$$

where $Q^\pi_{\Delta t}$ is defined in (9) and $V^\pi_{\Delta t}(s) = \mathbb{E}_{a\sim\pi}[Q^\pi_{\Delta t}(s,a)]$. As shown in (Jia and Zhou, 2023), instantaneous advantage function is given by

$$q^\pi(s,a) := r(s,a) + [\mathcal{L} V^\pi](s,a) - \beta V^\pi(s). \tag{11}$$

Accordingly, Q^π can be written as

$$Q^\pi(s,a) = r(s,a) + [\mathcal{L}V^\pi](s,a) - \beta V^\pi(s) + V^\pi(s). \tag{12}$$

Taking expectation over $a \sim \pi$ yields

$$\mathbb{E}_{a\sim\pi}[Q^\pi(s,a)] = r^\pi(s) + \mathcal{L}_{b^\pi,\Sigma^\pi} V^\pi(s) - \beta V^\pi(s) + V^\pi(s) = V^\pi(s),$$

where the last equality follows from the Bellman equation for V^π. Substituting $V^\pi(s) = \mathbb{E}_{a\sim\pi}[Q^\pi(s,a)]$ into (12) recovers the continuous-time Q-Bellman equation (10).

Third, we adopt the formulation in Definition 2, rather than (12), because it expresses the equation entirely in terms of the Q-function. This avoids the need to explicitly compute V^π and is essential for off-policy learning.

Next, we define the continuous-time optimal Bellman equation for $Q^*(s,a)$ as follows.

Definition 3 (Continuous-time optimal Bellman equation for $Q^*(s,a)$)**.**

$$Q^*(s,a) = r(s,a) + \mathcal{L}_{b,\Sigma}\big[\max_a Q^*(s,a)\big] + (1-\beta)\big[\max_a Q^*(s,a)\big]. \tag{13}$$

To verify that this equation is consistent with the standard RL relation $V^*(s) = \max_a Q^*(s,a)$, we provide a formal proof below. Define the operators

$$T_Q(Q) = r(s,a) + \mathcal{L}_{b,\Sigma}\big[\max_a Q(s,a)\big] + (1-\beta)\big[\max_a Q(s,a)\big], \tag{14}$$

$$T_V(V) = \max_a \Big(r(s,a) + [\mathcal{L}_{b,\Sigma}V](s,a) + (1-\beta)V\Big). \tag{15}$$

Let Q^* denote the fixed point of T_Q, i.e., $T_Q(Q^*) = Q^*$. Define $\hat{V}(s) = \max_a Q^*(s,a)$. Then

$$\begin{aligned} T_V(\hat{V}) &= \max_a \Big(r(s,a) + [\mathcal{L}_{b,\Sigma}\hat{V}](s,a) + (1-\beta)\hat{V}\Big) \\ &= \max_a \Big(r(s,a) + \mathcal{L}_{b,\Sigma}\big[\max_a Q^*(s,a)\big] + (1-\beta)\big[\max_a Q^*(s,a)\big]\Big) \\ &= \max_a Q^*(s,a) = \hat{V}. \end{aligned}$$

Hence, $\hat{V}$ is a fixed point of the operator T_V. Since V^*, the optimal value function, is also the unique fixed point of T_V (by the existence and uniqueness of the solution to the HJB equation), we conclude that $\hat{V} = V^*$. Consequently, the identity $V^*(s) = \max_a Q^*(s,a)$ holds.

We summarize several key properties of the continuous-time Q^π and Q^* as follows:

- The function $Q^\pi(s,a)$ admits the equivalent definition

$$Q^\pi(s,a) = \lim_{\Delta t\to 0}\left[\frac{Q^\pi_{\Delta t}(s,a) - V^\pi_{\Delta t}(s)}{\Delta t}\right] + V^\pi(s).$$

- The solution to the continuous-time Q-Bellman equation (2) is consistent with the value

function V^π in (2), in the sense that

$$\mathbb{E}_{a\sim\pi}[Q^\pi(s,a)] = V^\pi(s).$$

- The solution to the continuous-time optimal Q-Bellman equation (3) satisfies

$$\max_a Q^*(s,a) = V^*(s) = \max_\pi V^\pi(s) = \max_\pi \mathbb{E}_{a\sim\pi}[Q^\pi(s,a)].$$

3.2 Q-learning under Known Dynamics

In this section, we introduce an iterative algorithm to solve (13) for $Q^*(s,a)$ under linear bases. Let $\Phi(s) : \mathcal{S} \to \mathbb{R}^n$ be a set of basis functions. We approximate the solution by $\hat{Q}(s,a) = \Phi(s,a)^\top\theta^*$ using Galerkin method. The Galerkin approximation $\hat{Q}(s,a)$ satisfies,

$$\left\langle r(s,a) + \mathcal{L}_{b,\Sigma}\left(\max_a \left[\Phi(s,a)^\top\theta^*\right]\right) + (1-\beta)\max_a\left[\Phi(s,a)^\top\theta^*\right] - \Phi(s,a)^\top\theta^*, \Phi(s,a)\right\rangle = 0 \tag{16}$$

However, directly solving the above equation for θ^* is infeasible. We add a time dependence on $\theta(t)$, and derive an ODE for $\theta(t)$

$$\begin{aligned}\frac{d}{dt}\theta(t) =& \Big\langle r(s,a) + \mathcal{L}_{b,\Sigma}\Big(\max_a\left[\Phi(s,a)^\top\theta(t)\right]\Big) \\ &+ (1-\beta)\max_a\left[\Phi(s,a)^\top\theta(t)\right] - \Phi(s,a)^\top\theta(t),\ \Phi(s,a)\Big\rangle\end{aligned}$$

Discretize the time evolution, one has

$$\begin{aligned}\theta_{n+1} =& \theta_n + \alpha_n\Big\langle r(s,a) + \mathcal{L}_{b,\Sigma}\Big(\max_a[\Phi(s,a)^\top\theta_n]\Big) \\ &+(1-\beta)\max_a[\Phi(s,a)^\top\theta_n] - \Phi(s,a)^\top\theta_n,\ \Phi(s,a)\Big\rangle\end{aligned} \tag{17}$$

We establish the properties of the proposed algorithm in two steps. First, Theorem 3.2 shows that, under Assumption 2, the Galerkin solution θ^* defined in (16) provides a good approximation to the true value function $Q^*(s,a)$ defined in (13). Second, Theorem 3.3 shows that the iterates θ_n generated by Algorithm (17) converge to the Galerkin approximation θ^*.

Assumption 2. *(A1) The smallest eigenvalue λ_1 of the Gram matrix $G = \int \Phi\Phi^\top\mu dsda$ is positive. The largest eigenvalue λ_2 of the Gram matrix is bounded, and the bases $\|\Phi\|_\infty$ are bounded. In addition, the largest eigenvalue of Gram matrix G_∇, G_{∇^2} of $\nabla\Phi$, $\nabla^2\Phi$ are bounded. Let $c_3 := \max_{v\in\mathbf{S}^{n-1}} \frac{\|\max_a |v^\top\Phi|\|}{\|\Phi^\top v\|}$, we assume that*

$$c_\beta = 1 - c_3|1-\beta| > 0 \tag{18}$$

(A2) The functions $b(s,a) \in C^1(\mathbb{S}\times\mathbb{A}), \Sigma(s,a) \in C^2(\mathbb{S}\times\mathbb{A})$ satisfy:

$$\|\nabla b\|_\infty + \frac{1}{2}\|\nabla^2\Sigma\|_\infty \le \frac{c_\beta}{6c_3}, \quad \|b\|_\infty + \frac{1}{2}\|\nabla\Sigma\|_\infty \le \frac{c_\beta c_1}{6c_3}, \quad \frac{1}{2}\|\Sigma\|_\infty \le \frac{c_\beta c_2}{6c_3}.$$

where $c_1 := \min_{v\in\mathbf{S}^{n-1}} \frac{\|\Phi^\top v\|}{\|\nabla\Phi^\top v\|}, c_2 := \min_{v\in\mathbf{S}^{n-1}} \frac{\|\Phi^\top v\|}{\|\nabla^2\Phi^\top v\|}$

Several remarks on the assumptions are in order. Assumption (A1) constrains the choice of basis functions $\Phi(s)$ and the discount parameter β, whereas Assumption (A2) imposes conditions on the underlying dynamics.

In Assumption (A1), the requirement that the smallest eigenvalue of the Gram matrix be strictly positive is equivalent to linear independence of the basis functions in the weighted space $L^2(\mu)$. The condition $c_\beta > 0$ requires either a sufficiently small c_3 or the discount coefficient β sufficiently close to one.

The constant c_β is one of the key quantities governing both the assumptions on the dynamics and the performance of the algorithm. When c_β is larger, Assumption (A2) imposes weaker restrictions on the dynamics, allowing for more oscillatory systems. Later, in Theorems 3.2 and 3.3, we will show that both the approximation accuracy and convergence behavior improve when c_β is larger. Additional discussion will be provided after Theorem 3.3.

Next, we discuss the regime $\beta \in (0, 1)$. One observes a positive relationship between the discount coefficient β and the assumptions imposed on the dynamics. When β is larger, i.e., closer to 1, Assumption (A2) permits more oscillatory systems. In contrast, when β is closer to 0, the optimal control problem becomes less discounted, and Assumption (A2) requires stronger regularity conditions on the dynamics. This type of trade-off is also common in infinite-horizon discounted optimal control problems. However, our assumption requires β to remain within a moderate range and not be too far from 1. This assumption is mainly technical and may potentially be relaxed with a more refined analysis, which we leave for future work.

We next discuss the constant c_3. First, note that c_3 is always finite. Indeed, for any $v \in \mathbb{R}^d$ with $\|v\|_2 = 1$, $\|v^\top\Phi\|^2 = v^\top G v \geq \lambda_1$, where λ_1 is the smallest eigenvalue of G. Hence, $c_3 \leq \frac{\|\Phi\|_\infty}{\sqrt{\lambda_1}}$.

Next, consider the special case where the action space $\mathcal{A}$ is finite with $K = |\mathcal{A}|$ actions. Let $\mu(s) = \sum_{i=1}^K \mu(s, a_i)$ be the marginal distribution over states, and define the conditional distribution $\mu(a_i \mid s) = \frac{\mu(s,a_i)}{\mu(s)}$, which corresponds to the behavior policy generating the data. Define $\mu_0(s) = \min_i \mu(a_i \mid s)$. Then,

$$\left(\max_{a_i} |v^\top\Phi(s, a_i)|\right)^2 \leq \sum_i |v^\top\Phi(s, a_i)|^2 \leq \frac{1}{\mu_0(s)} \sum_i |v^\top\Phi(s, a_i)|\mu(a_i|s) \tag{19}$$

Therefore,

$$\begin{aligned}
\|\max_a |v^\top\Phi|\|^2 &= \int \left(\max_{a_i} |v^\top\Phi|\right)^2 \mu(s)ds \\
\leq &\frac{1}{\mu_0(s)} \int \sum_i |v^\top\Phi(s, a_i)|\mu(a_i|s)\mu(s)ds = \max_s \left(\frac{1}{\mu_0(s)}\right) \left\|v^\top\Phi\right\|^2,
\end{aligned}$$

which implies

$$c_3 \leq \sqrt{\max_s \left(\frac{1}{\mu_0(s)}\right)}. \tag{20}$$

Note that the RHS of (20) is minimized when the behavior policy is uniform over the action space, i.e., $\mu(a_i \mid s) = 1/K$, in which case $c_3 \leq \sqrt{K}$. This suggests that when the behavior policy explores the action space more uniformly, the proposed algorithm achieves faster convergence and better approximation accuracy in the presence of model error. Finally, we note that the bound $\sqrt{K}$ is generally loose because the first inequality in (19) can be highly conservative. In practice, the actual value of c_3 is often substantially smaller than $\sqrt{K}$ in this special case.

Assumption (A2) imposes smoothness and boundedness conditions on the drift b, diffusion Σ. Specifically, the bounds require that the magnitude and gradient magnitude of b and Σ are sufficiently small compared with λ_1, so that the Galerkin approximation remains stable. In addition, we note that the constants c_1, c_2, c_3 are always strictly positive under Assumption (A1). This is because $\left\|v^\top \Phi\right\| = v^\top G v \geq \lambda_1$ for $\|v\|_2 = 1$. Let λ_3 and λ_4 denote the largest eigenvalues of the matrices G_∇ and G_{∇^2}, respectively. Then one can further bound $c_1 \geq \frac{\lambda_1}{\lambda_3} > 0$, $c_2 \geq \frac{\lambda_1}{\lambda_4} > 0$. These bounds highlight that the admissible magnitudes of $\|b\|_{C^1}$ and $\|\Sigma\|_{C^2}$ depend on the sensitivity of the basis functions. When Φ is smooth and dominated by low-frequency components under $L^2(\mu)$, larger dynamics can be accommodated. In contrast, rapidly varying bases (with large gradients or curvature) require correspondingly smaller drift and diffusion to maintain stability.

It is worth emphasizing that our assumptions are primarily for theoretical guarantees and are not sharp. In the numerical experiments, we show that even when these conditions are violated, the algorithm still exhibits stable convergence in practice.

Before we present the theorem, we first prove a Lemma that will be frequently used in the theoretical guarantees.

Lemma 3.1. *Under Assumption 2, for any $Q = \Phi^\top \theta$ in the linear space spanned by Φ, one has the following upper bound,*

$$\left\|(1-\beta)Q + \mathcal{L}^*_{b,\Sigma} Q\right\| \leq \frac{1 - c_\beta/2}{c_3} \|Q\|, \quad \textit{where} \quad \mathcal{L}^*_{b,\Sigma} f = \nabla \cdot \left[bf + \frac{1}{2}\nabla \cdot (\Sigma f)\right].$$

Proof. First note that

$$\begin{aligned}
&\left\|(1-\beta)Q + \mathcal{L}^*_{b,\Sigma} Q\right\| \\
\leq &|1-\beta| \|Q\| + \left(\|\nabla b\|_\infty + \frac{1}{2}\left\|\nabla^2 \Sigma\right\|_\infty\right)\|Q\| + (\|b\|_\infty + \|\nabla \Sigma\|)\|\nabla Q\| + \frac{1}{2}\|\Sigma\|_\infty \left\|\nabla^2 Q\right\| \\
\leq &\frac{1-c_\beta}{c_3}\|Q\| + \frac{c_\beta}{6c_3}\left(\|Q\| + c_1\left\|\nabla \Phi^\top \theta\right\| + c_2\left\|\nabla^2 \Phi^\top \theta\right\|\right) \\
\leq &\frac{1-c_\beta}{c_3}\|Q\| + \frac{c_\beta}{2c_3}\|Q\| = \frac{1-c_\beta/2}{c_3}\|Q\|
\end{aligned}$$

where the first inequality is due to Cauchy–Schwarz inequality; the second inequaltiy is by Assumption (A2) and definition of c; the third inequality is because $Q = \theta^\top \Phi$ and for

$\forall 0 \neq \theta \in \mathbb{R}^n$, one has

$$c_1 \left\|\nabla \Phi^\top \theta\right\| = \min_{v \in \mathbf{S}^{n-1}} \frac{\left\|\Phi^\top v\right\|}{\left\|\nabla \Phi^\top v\right\|} \left\|\nabla \Phi^\top \theta\right\| \leq \left\|\theta^\top \Phi\right\|,$$
$$c_2 \left\|\nabla^2 \Phi^\top \theta\right\| = \min_{v \in \mathbf{S}^{n-1}} \frac{\left\|\Phi^\top v\right\|}{\left\|\nabla^2 \Phi^\top v\right\|} \left\|\nabla^2 \Phi^\top \theta\right\| \leq \left\|\theta^\top \Phi\right\|.$$

□

We first prove that the Galerkin approximation $\hat{Q}^*(s,a)$ is a good approximation to the true optimal value function $Q^*(s,a)$.

Theorem 3.2 (Galerkin solution accuracy)**.** *Under Assumption 2, the Galerkin solution* $\hat{Q}^*(s,a) = \Phi(s,a)^\top \theta^*$ *that satisfies* (16) *has an accuracy of*

$$\left\|\hat{Q}^* - Q^*\right\| \leq C_G \min_\theta \left\|Q^* - \Phi^\top \theta\right\|$$

where $C_G = \frac{4 - c_\beta}{c_\beta}$ *is a constant only depending on* Φ, β.

Several remarks on the above theorem are in order. First, when the optimal value function $Q^*(s,a)$ can be represented by the bases $\Phi(s,a)$, then the Galerkin solution is accurate. Second, one can view $\min_\theta \left\|Q^* - \Phi^\top \theta\right\|$ as the model error, that is the best approximation one can obtain in the space spanned by the bases Φ. The theorem shows that the Galerkin solution $\hat{Q}^*$ achieves, up to a multiplicative constant C_G, the smallest possible error among all linear combinations of Φ. Therefore, whenever the linear basis $\Phi(s,a)$ provides a good approximation to Q^*, the Galerkin solution inherits this accuracy and is guaranteed to be near-optimal within the chosen function class.

Note that, by the definition in (18), one has $c_\beta < 1$, which implies $C_G > 0$. In the presence of model approximation error, the accuracy of the Galerkin solution improves as c_β becomes larger. Based on the discussion following Assumption 2, this implies that the Galerkin approximation becomes more accurate when β is larger or when the behavior policy explores the action space more uniformly.

Proof. We divide the error $e(s,a) = Q^*(s,a) - \hat{Q}^*(s,a)$ into two parts,

$$e(s,a) = e_G(s,a) + e_P(s,a), \quad e_P = Q^* - \Phi^\top u, \quad e_G = \Phi^\top v, \quad v = u - \theta^* \tag{21}$$

where $u, v \in \mathbb{R}^d, u = \arg\min_\theta \left\|Q^* - \Phi^\top \theta\right\|$. Here e_G represents the Galerkin error, and e_P represents the model error.

Taking inner product of the continuous-time Q-function (3) with e_G, one has

$$\left\langle r + \mathcal{L}_{b,\Sigma}(\max_a Q^*) + (1-\beta) \max_a Q^*, e_G \right\rangle = \langle Q^*, e_G \rangle$$

Taking inner product of v with the Galerkin equation (16), one has

$$\left\langle r + \mathcal{L}_{b,\Sigma}(\max_a \hat{Q}^*) + (1-\beta) \max_a \hat{Q}^*, e_G \right\rangle = \left\langle \hat{Q}^*, e_G \right\rangle$$

subtracting the two equations gives,

$$\left\langle \mathcal{L}_{b,\Sigma}(\max_a Q^* - \max_a \hat{Q}^*) + (1-\beta)(\max_a Q^* - \max_a \hat{Q}^*), e_G \right\rangle = \|e_G\|^2 + \langle e_P, e_G \rangle$$

Now the LHS can be bounded by

$$\begin{aligned} LHS =& \left\langle \max_a Q^* - \max_a \hat{Q}^*, (1-\beta)e_G + \mathcal{L}^*_{b,\Sigma} e_G \right\rangle \leq \frac{1-c_\beta/2}{c_3} \left\| \max_a Q^* - \max_a \hat{Q}^* \right\| \|e_G\| \\ \leq& \frac{1-c_\beta/2}{c_3} \left\| \max_a |Q^* - \hat{Q}^*| \right\| \|e_G\| = \frac{1-c_\beta/2}{c_3} \left\| \max_a |e_G + e_P| \right\| \|e_G\| \\ \leq& \frac{1-c_\beta/2}{c_3} \left(\left\| \max_a |e_G| \right\| \|e_G\| + \left\| \max_a |e_P| \right\| \|e_G\| \right) \\ \leq& (1-c_\beta/2) \left(\|e_G\|^2 + \|e_P\| \|e_G\| \right) \end{aligned}$$

where the first equality is due to integration by parts, and the first inequality is by Lemma 3.1. The last inequality is because of Assumption 2/(A1), for any $e(s,a) = e^\top \Phi(s,a)$ with $\|e\|_2 \neq 0$,

$$\left\| \max_a |e| \right\| = \frac{\|\max_a |e|\|}{\|e\|} \|e\| \leq \max_{v \in \mathbf{S}^{n-1}} \frac{\|\max_a |v^\top \Phi|\|}{\|\Phi^\top v\|} \|e\| = c_3 \|e\| . \tag{22}$$

Therefore, one has

$$\frac{c_\beta}{2} \|e_G\|^2 \leq (1-c_\beta/2) \|e_P\| \|e_G\| + \langle e_P, e_G \rangle \leq (2-c_\beta/2) \|e_P\| \|e_G\|$$

which implies

$$\|e_G\| \leq \frac{4-c_\beta}{c_\beta} \|e_P\| .$$

One completes the proof by inserting the definition of e_P in (21). □

The convergence guarantee is derived in the following theorem.

Theorem 3.3 (Convergence of Q-learning under known dynamics)**.** *Under Assumption 2, the Q-learning iteration defined in (17) with learning rate $\alpha_n = \alpha \in (0, \frac{c_\beta \lambda_1}{4\lambda_2^2})$ will converges to the Galerkin solution θ^* defined in (16) in the following rate,*

$$\|\theta_n - \theta^*\|_2 \leq (1 - \alpha c_\beta \lambda_1 + 4\alpha^2 \lambda_2^2) \|\theta_0 - \theta^*\|_2$$

By setting the learning rate $\alpha = \frac{c_\beta \lambda_1}{8\lambda_2^2}$, the error decays as below:

$$\|\theta_n - \theta^*\|_2 \leq \left(\sqrt{1 - \left(\frac{c_\beta \lambda_1}{4\lambda_2} \right)^2} \right)^n \|\theta_0 - \theta^*\|_2.$$

Several remarks on the theorem are in order. The optimal convergence rate depends explicitly on the spectrum of the Gram matrix through λ_1 and λ_2, as well as the key parameter

c_β. In particular, better-conditioned bases lead to faster convergence. In the special case where the basis functions are orthonormal under $L^2(\mu)$, the Gram matrix becomes the identity, so $\lambda_1 = \lambda_2 = 1$, yielding the optimal rate. In addition, a larger c_β also leads to faster convergence. Based on the discussion following Assumption 2, this implies that the algorithm converges faster when β is larger or when the behavior policy explores the action space more uniformly.

We emphasize, however, that this convergence rate is derived as an upper bound and may not be sharp. In practice, the actual convergence behavior can be significantly faster, depending on additional structure of the problem and the interaction between the dynamics and the chosen basis.

Proof. Define

$$\mathcal{H}(\theta) = \left\langle \Phi^T\theta - (\mathcal{L}_{b,\Sigma} + (1-\beta)) \max_a \Phi^T\theta - r, \Phi \right\rangle. \tag{23}$$

Monotonicity. We first show that $\mathcal{H}$ is strongly monotone. Let $Q = \theta^\top\Phi, P = \eta^\top\Phi$, expanding the left-hand side, we have

$$\begin{aligned}
&(\mathcal{H}(\theta) - \mathcal{H}(\eta))^\top (\theta - \eta) \\
=&\|Q-P\|^2 - (1-\beta)\langle \max_a Q - \max_a P, Q - P\rangle - \left\langle (b\nabla + \frac{1}{2}\Sigma : \nabla^2)(\max_a Q - \max_a P), Q - P \right\rangle \\
=&\|Q-P\|^2 - \langle \max_a Q - \max_a P, ((1-\beta)(Q-P) + \mathcal{L}^*_{b,\Sigma}(Q-P)\ \rangle \\
\geq&\|Q-P\|^2 - \frac{1-c_\beta/2}{c_3} \left\| \max_a Q - \max_a P \right\| \|Q-P\| \geq \frac{c_\beta}{2}\|Q-P\|^2
\end{aligned}$$

where the first inequality is by Lemma 3.1, and the second inequality is obtained by applying (22),

$$\left\| \max_a Q - \max_a P \right\| \leq \left\| \max_a |Q-P| \right\| \leq c_3 \|Q-P\|$$

Again by Assumption (A1), one has

$$\|Q-P\|^2 = (\theta-\eta)^\top \left[\int \Phi\Phi^\top \mu(s,a) ds da \right] (\theta-\eta) \geq \lambda_1 \|\theta-\eta\|_2^2$$

Therefore, one has

$$(\mathcal{H}(\theta) - \mathcal{H}(\eta))^\top (\theta-\eta) \geq \frac{c_\beta \lambda_1}{2} \|\theta-\eta\|_2^2.$$

Lipschitz Continuity of $\mathcal{H}$. We verify that $\mathcal{H}$ is Lipschitz. Observe

$$\begin{aligned}
\|\mathcal{H}(\theta) - \mathcal{H}(\eta)\| \leq& \left\| \langle \Phi^\top(\theta-\eta), \Phi \rangle \right\| + \left\| \max_a Q - \max_a P \right\| \left\| (1-\beta)\Phi + \mathcal{L}^*_{b,\Sigma}\Phi \right\| \\
\leq& G(\theta-\eta) + \frac{1-c_\beta}{2} \|Q-P\| \|\Phi\| \leq 2\lambda_2 \|\theta-\eta\|_2
\end{aligned}$$

where Lemma 3.1 and (22) is applied to obtained the second inequality, and Assumption 2/(A1) is used in the third inequality. Then one has,

$$\|\mathcal{H}(\theta) - \mathcal{H}(\eta)\| \leq 2\lambda_2 \|\theta - \eta\|_2$$

Contraction Property. For fixed learning rate $\alpha_k = \alpha$, since the Galerkin iteration update is

$$\theta_{k+1} = \theta_k - \alpha_k \mathcal{H}(\theta_k),$$

and $\mathcal{H}(\theta^*) = 0$, one has

$$\begin{aligned}&\|\theta_{n+1} - \theta^*\|_2^2 = \|\theta_n - \theta^* - \alpha(\mathcal{H}(\theta_n) - \mathcal{H}(\theta^*))\|_2^2 \\ \leq &\|\theta_n - \theta^*\|_2^2 - 2\alpha(\mathcal{H}(\theta_n) - \mathcal{H}(\theta^*))^\top(\theta_n - \theta^*) + \alpha^2\|\mathcal{H}(\theta_n) - \mathcal{H}(\theta^*)\|_2^2\end{aligned}$$

By Monotonicity and Lipschitz continuity of $\mathcal{H}$, we have

$$\|\theta_{n+1} - \theta^*\|_2^2 \leq (1 - 2m\alpha + \alpha^2 K^2)\|\theta_n - \theta^*\|_2^2, \quad m = \frac{\lambda_1 c_\beta}{2}, \quad K = 2\lambda_2.$$

First note that since $K > m$, one has $1 - 2\alpha m + \alpha^2 K^2 > 0$. Second, one chooses $0 < \alpha < \frac{2m}{K^2}$ then the mapping becomes a contraction. Lastly, by setting $\alpha = m/K^2$, the convergence rate is optimized as

$$\|\theta_n - \theta^*\|_2 \leq \left(\sqrt{1 - \frac{m^2}{K^2}}\right)^n \|\theta_0 - \theta^*\|_2$$

. □

3.3 Q-learning under unknown dynamics

In the previous section, we assumed that the continuous dynamics b, σ are known. In this section, we assume b, σ are unknown. In Section 3.3.1, we introduce PhiBE to naturally applly the algorithm and theoretical guarantees when only the discrete-time transition dynamics are known. In 3.3.2, we describe how to implement the update rule (17) in a data-driven manner when the transition dynamics is unknown and only the state-action-next-state-reward tuples $\{(s_i, a_i, s_i', r_i)\}_{i=1}^n$ as defined in (5) are available.

3.3.1 PhiBE setting

When only the discrete-time transition dynamics $\rho_{\Delta t}(s'|s, a)$ are available, one can replace the unknown drift and diffusion b, Σ in the equation (13) for Q^* with the approximated $\hat{b}$ and $\hat{\Sigma}$ according to (7), and one ends up with the Optimal-PhiBE-Q equation

Definition 4 (Optimal-PhiBE-Q)**.**

$$\hat{Q}^*(s, a) = r(s, a) + \mathcal{L}_{\hat{b},\hat{\Sigma}}\big[\max_a \hat{Q}^*(s, a)\big] + (1 - \beta)\big[\max_a \hat{Q}^*(s, a)\big], \tag{24}$$

where $\hat{b}, \hat{\Sigma}$ *are defined in* (7)*.*

Substituting $(\hat{b}, \hat{\Sigma})$ into (14) yields a corresponding operator, whose fixed point corresponds to the optimal PhiBE solution to (6). Solving the Optimal-PhiBE-Q equation (4), the same algorithm (17) applies with the approximated $\hat{b}, \hat{\Sigma}$. Therefore as long as $\hat{b}, \hat{\Sigma}$ satisfies the Assumption (A2), the corresponding algorithm

$$\begin{aligned}\theta_{n+1} =&\theta_n + \alpha_n \Big\langle r(s,a) + \mathcal{L}_{\hat{b},\hat{\Sigma}}\Big(\max_a [\Phi(s,a)^\top \theta_n]\Big) \\ &+ (1-\beta)\max_a [\Phi(s,a)^\top \theta_n] - \Phi(s,a)^\top \theta_n,\ \Phi(s,a)\Big\rangle\end{aligned} \tag{25}$$

recovers the same limiting behavior as established in Theorem 3.2 and Theorem 3.3.

Corollary 1. *Under Assumption 2/(A1), and for $\forall \epsilon > 0$, if*

$$\|\nabla b\|_\infty + \frac{1}{2}\|\nabla^2 \Sigma\|_\infty \le \frac{1}{6c_3} - \epsilon, \quad \|b\|_\infty + \frac{1}{2}\|\nabla \Sigma\|_\infty \le \frac{c_1}{6c_3} - \epsilon, \quad \frac{1}{2}\|\Sigma\|_\infty \le \frac{c_2}{6c_2} - \epsilon, \tag{26}$$

and Δt small enough, then the Galerkin solution $\tilde{Q}^*(s,a) = \Phi(s,a)^\top \tilde{\theta}^*$ for Optimal-PhiBE-Q,

$$\Big\langle r(s,a) + \mathcal{L}_{\hat{b},\hat{\Sigma}}\Big(\max_a \Big[\Phi(s,a)^\top \tilde{\theta}^*\Big]\Big) + (1-\beta)\max_a \Big[\Phi(s,a)^\top \tilde{\theta}^*\Big] - \Phi(s,a)^\top \tilde{\theta}^*, \Phi(s,a)\Big\rangle = 0 \tag{27}$$

has an accuracy of

$$\Big\|\tilde{Q}^*(s,a) - \hat{Q}^*(s,a)\Big\| \le C_G \min_\theta \Big\|\hat{Q}^*(s,a) - \Phi^\top \theta\Big\|_\infty$$

where C_G is the same constant in Theorem 3.2. In addition, with the same learning rate as in Theorem 3.3, the iterative algorithm 25 will converges to $\tilde{\theta}^*$ in the same rate as in Theorem 3.3.

Proof. As proved in (Zhu et al., 2025), $\Big\|\nabla^k(\hat{b} - b)\Big\|_\infty \le C\Delta t$ with $k = 0, 1$, $\Big\|\nabla^k(\Sigma - \hat{\Sigma})\Big\|_\infty \le C\Delta t$ with $k = 0, 1, 2$. Therefore, for sufficiently small Δt, one has

$$\|\nabla b - \nabla \hat{b}\|_\infty + \frac{1}{2}\|\nabla^2 \Sigma - \nabla^2 \hat{\Sigma}\|_\infty \le \epsilon$$

then by the assumption (26), one has

$$\|\nabla \hat{b}\|_\infty + \frac{1}{2}\|\nabla^2 \hat{\Sigma}\|_\infty \le \|\nabla b\|_\infty + \frac{1}{2}\|\nabla^2 \Sigma\|_\infty + \|\nabla b - \nabla \hat{b}\|_\infty + \frac{1}{2}\|\nabla^2 \Sigma - \nabla^2 \hat{\Sigma}\|_\infty \le \frac{1}{6c_3}.$$

Similary, one can prove with the same Δt,

$$\|\hat{b}\|_\infty + \frac{1}{2}\|\nabla \hat{\Sigma}\|_\infty \le \frac{c_1}{6c_3}, \quad \frac{1}{2}\|\hat{\Sigma}\|_\infty \le \frac{c_2}{6c_2} - \epsilon,$$

Therefore, the approximated $\hat{b}, \hat{\Sigma}$ sastisfies Assumption 2/(A2), then the guarantees in Theorems 16 and 27 follows.

□

3.3.2 Data-driven algorithm

When only discrete-time off-policy data (5) is available, our goal is to approximate the optimal action-value function $Q^*(s,a)$.

Following the algorithm (25) introduced in last section, we approximate the optimal-Q function $Q^*(s,a)$ using the bases $\Phi(s,a) = (\phi_1, \dots, \phi_n)^\top$. In each iteration, given $Q_n(s,a) = \Phi(s,a)^\top \theta_n$, one first computes

$$V_n(s) = \max_a Q_n(s,a).$$

Then a mini-batch $\mathcal{B}_n \subset \mathcal{B}$ is sampled, and for each tuple (s_i, a_i, r_i) in the mini-batch, the TD-error is computed as

$$\delta(s_i, s_i', a_i, r_i) = r_i + \mathcal{L}_{b_i,\Sigma_i} V_n(s_i) + (1-\beta) V_n(s_i) - Q_n(s,a), \tag{28}$$

which serves as an unbiased estimate of the PhiBE residual. Here $\mathcal{L}_{b,\Sigma}$ is defined in (4), and

$$b_i = \frac{s_i' - s_i}{\Delta t}, \quad \hat{\Sigma}_i = \frac{(s_i' - s_i)(s_i' - s_i)^\top}{\Delta t}.$$

Finally, the parameter is updated via

$$\theta_{n+1} = \theta_n + \frac{\alpha_n}{|\mathcal{B}_n|} \sum_{(s_i, s_i', a_i, r_i) \in \mathcal{B}_n} \Phi(s_i, a_i)\, \delta(s_i, s_i', a_i, r_i), \tag{29}$$

This procedure is repeated until a satisfactory approximation of the Galerkin solution for $Q^*(s,a)$ is obtained. A rigorous analysis of the sample complexity is left for future work. The algorithm is presented in Algorithm 1.

Algorithm 1 PhiBE Q-learning Method

Input: Batch size $|\mathcal{B}_n|$, initial value θ_0, discrete time step Δt, discount β, state-next-state-reward-action tuples $\mathcal{B} = \{(s_i, s_i', r_i, a_i)\}_{i=1}^N$, basis $\Phi(s,a) = (\phi_1, \dots, \phi_n)^\top$, steps N_{iter}, step-sizes $\{\alpha_n\}$

Output: Estimated parameter $\theta_{N_{\text{iter}}}$ and Q-function $\hat{Q}(s,a) = \theta^\top \Phi(s,a)$

$\theta \leftarrow \theta_0$

for $n = 0, \dots, N_{\text{iter}} - 1$ **do**

Compute the optimal state-value function $V(s) \leftarrow \max_a \Phi(s,a)^\top \theta$

Sample a mini-batch $\mathcal{B}_n \subset \mathcal{B}$, and compute PhiBE-Q error for each $(s_i, s_i', a_i, r_i) \in \mathcal{B}_n$

$$\delta(s_i, s_i', a_i, r_i) = \frac{s_i' - s_i}{\Delta t} \cdot \nabla_s V(s_i) + \tfrac{1}{2\Delta t}(s_i' - s_i)(s_i' - s_i)^\top : \nabla_s^2 V(s_i)$$

Update parameter: $\theta \leftarrow \theta + \frac{\alpha_n}{|\mathcal{B}_n|} \sum_{(s_i, s_i', a_i, r_i) \in \mathcal{B}_n} \Phi(s_i, a_i)\, \delta(s_i, a_i, r_i)$

end for

return $\hat{Q}(s,a) = \theta_{N_{\text{iter}}}^\top \Phi(s,a)$

Several remarks on the Algorithm are in order.

First, the algorithm can be viewed as a continuous-time analogue of classical Q-learning (Watkins and Dayan, 1992). The classical approach constructs the TD error as the residual of the discrete-time Bellman equation (8), whereas our method is based on the PhiBE-Q equation (24).

Second, the algorithm naturally extends to nonlinear function approximation, such as Deep Neural Networks (DNNs). One constructs the same TD error as in (28) using the parametric form $Q(s, a; \theta_n)$, and updates the parameters via

$$\theta_{n+1} = \theta_n + \alpha_n \, \delta(s_i, s_i', a_i, r_i) \nabla_\theta Q(s_i, a_i; \theta_n).$$

As in classical RL, convergence guarantees are generally unavailable under nonlinear function approximation. In practice, stability can be improved by decoupling the target from the online network. For example, following (Mnih et al., 2015), one may use a target network and define $V_n = \max_a Q(s, a; \theta^-)$, where θ^- denotes parameters from a delayed copy of the Q-network.

Third, with the continuous-time formulations of $Q^*(s, a)$ in (11) and $Q^\pi(s, a)$ in (2), standard off-policy algorithms DQN (Mnih et al., 2015) including Q-learning (Watkins and Dayan, 1992), DQN (Mnih et al., 2015), TD3-BC (Fujimoto and Gu, 2021), CQL (Kumar et al., 2020), etc can be directly adapted to solve continuous-time RL problems.

Finally, we would like to mention that this algorithm is typically applied in settings where $\max_a \Phi(s, a)^\top \theta$ can be computed efficiently. The numerical experiment in Section 4.1 provides one such example. For discrete action spaces, $\max_a \Phi(s, a)^\top \theta$ is easy to evaluate; however, $V_n(s)$ is not necessarily differentiable. To address this, one may approximate the greedy action $a^*(s) = \arg\max_a Q(s, a)$ using a softmax policy $\pi(a \mid s) \propto \exp\big(Q(s, a)/\lambda\big)$, where λ controls the degree of exploration. This leads to a softmax variant of PhiBE-Q-learning. Alternatively, one can approximate $\partial_{s_k} V(s_i) \approx \frac{1}{h}(V(s_i + he_k) - V(s_i))$, where e_k denotes the kth standard basis vector in the state space and $h > 0$ is a small finite-difference step size. Provided that $h < \Delta t$, the additional approximation error introduced by this finite-difference step does not dominate the overall discretization error.

4 Experiments

4.1 LQR with stochastic dynamics

We consider the linear dynamics with quadratic reward setting,

$$b(s, a) = As + Ba, \quad \sigma(s, a) = \sigma, \quad r(s, a) = s^\top Q s + a^\top R a,$$

The system parameters are set as $A = B = 0.1$, $Q = R = -1$, and the discount factor $\beta = 0.1$. In the noise-free case, $\sigma = 0$, whereas in the noisy case, $\sigma = 0.1$.

To generate the dataset, we simulate I independent trajectories $\{(s_i, s_i', r_i, a_i)\}_{i=1}^N$. Here, $s \sim U(-1, 1), a \sim U(-1, 1)$ and s' is the resulting next state after applying action a,

$$\rho(s' \mid s, a) \sim \mathcal{N}\left(e^{A\Delta t} s + \frac{B}{A}\big(e^{A\Delta t} - 1\big)\, a,\ \frac{\sigma^2}{2A}\big(e^{2A\Delta t} - 1\big)\right).$$

.

We represent $Q_\theta(s,a) = \theta^\top \Phi(s,a)$ using quadratic features $\Phi(s,a) = (s^2, a^2, 1)$, and initialize the parameter θ_0 randomly. Note that in LQR, the corresponding greedy value function admits the explicit representation

$$\max_a Q_\theta(s,a) = \theta_m^\top \phi(s), \quad \theta_m = \left(\theta_1 - \frac{\theta_2^2}{\theta_3},\, 0,\, \theta_3\right)^\top.$$

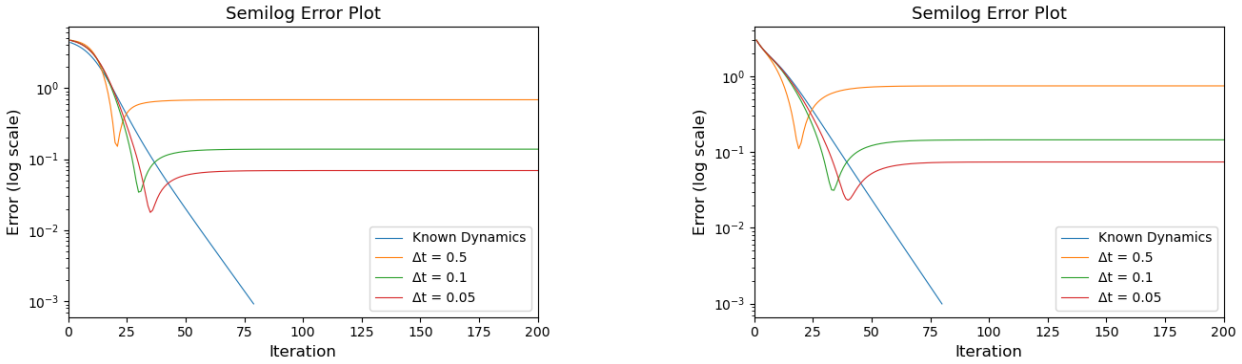
The step size in all plots is set to $\alpha_n = 0.1$. Figure 1 shows the evolution of the error $|\theta_n - \theta^*|$ for PhiBE under known discrete-time dynamics. We implement the algorithm in (25), where $\hat{b}$, $\hat{\Sigma}$, and the inner products are computed explicitly. The purpose of this experiment is to demonstrate that PhiBE provides an accurate approximation to the continuous-time optimal control problem and to validate our theoretical guarantees.

Figure 2 shows the evolution of the error $|\theta_n - \theta^*|$ under discrete-time data, using Algorithm 1. This experiment demonstrates that the proposed data-driven algorithm performs comparably to the algorithm with known discrete-time dynamics.

At each iteration, we sample a mini-batch from the dataset, compute the Optimal-PhiBE-Q residual defined in (28) at θ_n

$$\delta(s,a,r) = r + (\widehat{\mathcal{L}} \max Q_{\theta_n})(s) + (1-\beta) \max Q_{\theta_n}(s) - Q_{\theta_n},$$

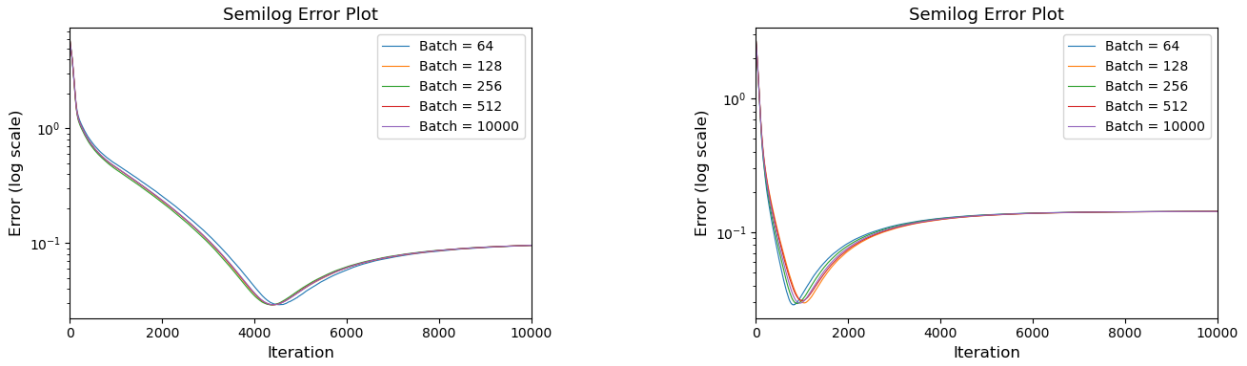
and update θ_n according to the PhiBE-Q-learning rule (29).



(a) Deterministic dynamics

(b) Stochastic dynamics

Figure 1: The evolution of PhiBE Q-learning under known discrete-time dynamics. Subfigure (a) shows the deterministic setting ($\sigma = 0$), while subfigure (b) shows the stochastic setting ($\sigma = 0.2$). $\beta = 1$ in the deterministic setting, $\beta = 2$ in the stochastic setting. In each subfigure, different curves correspond to the exact known-dynamics update and its corresponding PhiBE approximation using different time steps Δt.



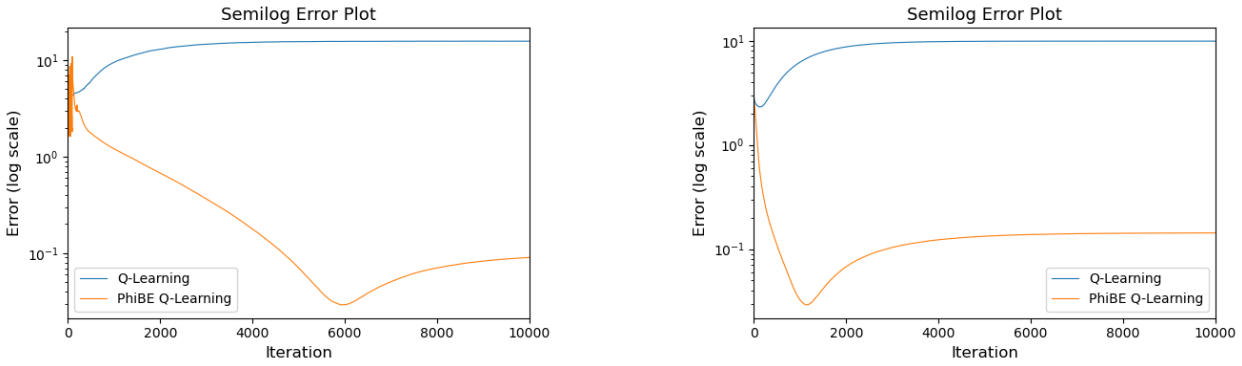
(a) Deterministic dynamics

(b) Stochastic dynamics

Figure 2: The evolution of PhiBE Q-learning when only discrete trajectory off-policy data are available with $\Delta t = 0.1$. Subfigure (a) shows the deterministic setting ($\sigma = 0$), while subfigure (b) shows the stochastic setting ($\sigma = 0.2$). In each subfigure, different curves correspond to different batch size used in each iteration. $\beta = 1$ in the deterministic setting, $\beta = 2$ in the stochastic setting.

For the known-dynamics setting, the results in Figure 1 show that PhiBE Q-learning closely tracks the exact discrete-time Q-iteration across both deterministic and stochastic dynamics. When the time step Δt decreases, the PhiBE approximation becomes more accurate. In the stochastic case the long-term convergence trend remains nearly identical to the deterministic counterpart. The algorithm ultimately reaches the same accuracy level, confirming that the PhiBE Q-learning update is stable and robust to moderate diffusion noise in the transition dynamics.

For the off-policy data setting in Figure 2, PhiBE Q-learning successfully recovers the Q-function from sample trajectories, and its convergence behavior is strongly influenced by the batch size. Larger batches lead to smoother error decay and significantly faster convergence, while very small batches introduce higher variance and slower stabilization. Comparing the deterministic and stochastic environments, we observe that stochasticity causes oscillations during the transient phase. However, by adopting a decreasing learning rate in later iterations, the method avoids persistent oscillation and converges to a similar final error plateau. These results demonstrate that PhiBE Q-learning remains effective even with off-policy data and stochastic dynamics, achieving reliable convergence across different sampling regimes.



(a) Deterministic dynamics

(b) Stochastic dynamics

Figure 3: The evolution of PhiBE Q-learning when only discrete trajectory off-policy data are available compared with RL Q-Learning under same batch size = 64. Subfigure (a) shows the deterministic setting ($\sigma = 0$), while subfigure (b) shows the stochastic setting ($\sigma = 0.2$). In each subfigure, different curves correspond to whether the method is based on PhiBE. We set $\Delta t = 0.1$, $\beta = 2$ in this setting.

To further demonstrate the effectiveness of our method, we also evaluate the standard Q-Learning algorithm using the same data and experimental configuration. As shown in Figure 3, the Q-Learning method iterates converge with a larger error. Therefore, the experimental results clearly confirm that our approach is significantly more robust than the standard Q-learning in the continuous setting.

5 Conclusion

In this work, we propose a continuous-time Q function, enabling off-policy data usage in continuous-time RL settings. By introducing an time-evolutionary equation and leveraging the Galerkin projection, we develop a stable iterative scheme for solving continuous-time optimal control problems in a model-free way.

We further establish convergence guarantees for the proposed algorithms under linear function approximation. The resulting stability conditions are independent of Δt, allowing our method to learn robustly from discrete trajectory data. Numerical experiments demonstrate the effectiveness of the proposed algorithms and highlight their superior convergence behavior compared to standard Q-learning, which may diverge in similar continuous-time settings.

Several promising directions remain open for future investigation. First, from a theoretical perspective, some of the assumptions used to establish the current guarantees may potentially be relaxed. While this paper focuses on convergence and convergence rate analysis, establishing theoretical sample complexity bounds is an important next step. Second, extending our methodology to large-scale problems via deep neural approximators presents both opportunities and challenges in expressiveness and stability. Third, incorporating exploration strategies and safety constraints would broaden applicability in real-world control systems.

Overall, this work provides a principled and data-driven foundation for continuous-time reinforcement learning for off-policy data, and we believe the proposed framework will

facilitate further developments toward scalable and reliable continuous-time decision-making.

References

Alekh Agarwal, Sham Kakade, and Lin F. Yang. Model-based reinforcement learning with a generative model is minimax optimal. In Jacob Abernethy and Shivani Agarwal, editors, Proceedings of Thirty Third Conference on Learning Theory, volume 125 of Proceedings of Machine Learning Research, pages 67–83. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/agarwal20b.html.

Leemon 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.

Tadej Battelino, Thomas Danne, Richard M Bergenstal, Stephanie A Amiel, Roy Beck, Torben Biester, Emanuele Bosi, Bruce A Buckingham, William T Cefalu, Kelly L Close, et al. Clinical targets for continuous glucose monitoring data interpretation: recommendations from the international consensus on time in range. Diabetes care, 42(8):1593–1603, 2019.

Dimitri Bertsekas. Dynamic programming and optimal control: Volume I, volume 4. Athena scientific, 2012.

Claudio Cobelli, Chiara Dalla Man, Giovanni Sparacino, Lalo Magni, Giuseppe De Nicolao, and Boris P Kovatchev. Diabetes: models, signals, and control. IEEE reviews in biomedical engineering, 2:54–96, 2009.

Harry Emerson, Matthew Guy, and Ryan McConville. Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes. Journal of Biomedical Informatics, 142:104376, 2023.

Xiaobing Feng, Roland Glowinski, and Michael Neilan. Recent developments in numerical methods for fully nonlinear second order partial differential equations. siam REVIEW, 55 (2):205–267, 2013.

Wendell H. Fleming and H. Mete Soner. Controlled Markov processes and viscosity solutions, volume 25 of Stochastic Modelling and Applied Probability. Springer, second edition, 2006.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. Advances in neural information processing systems, 34:20132–20145, 2021.

Yanwei Jia and Xun Yu Zhou. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. Journal of Machine Learning Research, 23(154): 1–55, 2022a.

Yanwei Jia and Xun Yu 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.

Yanwei Jia and Xun Yu Zhou. q-learning in continuous time. Journal of Machine Learning Research, 24(161):1–61, 2023.

Rushikesh Kamalapurkar, Patrick Walters, and Warren E Dixon. Model-based reinforcement learning for approximate optimal regulation. Automatica, 64:94–104, 2016.

Jeongho Kim and Insoon Yang. Hamilton-jacobi-bellman equations for q-learning in continuous time. In Learning for Dynamics and Control, pages 739–748. PMLR, 2020.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. Advances in neural information processing systems, 33: 1179–1191, 2020.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. Off-policy learning in two-stage recommender systems. In Proceedings of The Web Conference 2020, pages 463–473, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. nature, 518(7540):529–533, 2015.

Wenlong Mou and Yuhua Zhu. On bellman equations for continuous-time policy evaluation: discretization and approximation. arXiv preprint arXiv:2407.05966, 2024.

Michael Muehlebach, Zhiyu He, and Michael I Jordan. The sample complexity of online reinforcement learning: A multi-model perspective. arXiv preprint arXiv:2501.15910, 2025.

Huyên Pham. Continuous time stochastic control and optimization with financial applications, volume 61 of Stochastic modeling and applied probability. Springer-Verlag, New York, 2009.

Hayden Schaeffer and Thomas Y Hou. An accelerated method for nonlinear elliptic pde. Journal of Scientific Computing, 69(2):556–580, 2016.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. nature, 529(7587):484–489, 2016.

Richard S Sutton, Andrew G Barto, et al. Reinforcement learning: An introduction, volume 1. MIT press Cambridge, 1998.

Lukasz Szpruch, Tanut Treetanthiploet, and Yufei Zhang. Optimal scheduling of entropy regularizer for continuous-time linear-quadratic reinforcement learning. SIAM Journal on Control and Optimization, 62(1):135–166, 2024.

Corentin Tallec, Léonard Blier, and Yann Ollivier. Making deep q-learning methods robust to time discretization. In International Conference on Machine Learning, pages 6096–6104. PMLR, 2019.

Haoran Wang and Xun Yu Zhou. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. Mathematical Finance, 30(4):1273–1308, 2020.

Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. Journal of Machine Learning Research, 21(198):1–34, 2020.

Christopher JCH Watkins and Peter Dayan. Q-learning. Machine learning, 8(3):279–292, 1992.

Xiong Yang, Derong Liu, and Ding Wang. Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints. International Journal of Control, 87(3):553–566, 2014.

Cagatay Yildiz, Markus Heinonen, and Harri Lähdesmäki. Continuous-time model-based reinforcement learning. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 12009–12018. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/yildiz21a.html.

Jiongmin Yong and Xun Yu Zhou. Stochastic controls – Hamiltonian systems and HJB equations, volume 43 of Applications of Mathematics (New York). Springer-Verlag, New York, 1999.

Runze Zhao, Yue Yu, Adams Yiyue Zhu, Chen Yang, and Dongruo Zhou. Sample and computationally efficient continuous-time reinforcement learning with general function approximation. arXiv preprint arXiv:2505.14821, 2025.

Yuhua Zhu. Phibe: A pde-based bellman equation for continuous time policy evaluation. arXiv preprint arXiv:2405.12535, 2024.

Yuhua Zhu, Yuming Zhang, and Haoyu Zhang. Optimal-phibe: A pde-based model-free framework for continuous-time reinforcement learning. arXiv preprint arXiv:2506.05208, 2025.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom Brown, Alec Radford, Dario Amodei, and Paul F. Christiano. Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593, 2019.