DECOUPLED CONTINUOUS-TIME REINFORCEMENT LEARNING VIA HAMILTONIAN FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world control problems, ranging from finance to robotics, evolve in continuous time with non-uniform, event-driven decisions. Standard discrete-time reinforcement learning (RL), based on fixed-step Bellman updates, struggles in this setting: as time gaps shrink, the Q-function collapses to the value function V, eliminating action ranking. Existing continuous-time methods reintroduce action information via an advantage-rate function q. However, they enforce optimality through complicated martingale losses or orthogonality constraints, which are sensitive to the choice of test processes. These approaches entangle V and q into a large, complex optimization problem that is difficult to train reliably. To address these limitations, we propose a novel decoupled continuous-time actor-critic algorithm with alternating updates: q is learned from diffusion generators on V, and V is updated via a Hamiltonian-based value flow that remains informative under infinitesimal time steps, where standard max/softmax backups fail. Theoretically, we prove rigorous convergence via new probabilistic arguments, sidestepping the challenge that generator-based Hamiltonians lack Bellman-style contraction under the sup-norm. Empirically, our method outperforms prior continuous-time and leading discrete-time baselines across challenging continuous-control benchmarks and a real-world trading task, achieving 21% profit over a single quarter—nearly doubling the second-best method.

1 INTRODUCTION

Many learning and control problems in scientific computing, robotics, and finance evolve in continuous time with continuous state and action spaces. In markets, decisions are often made only when new signals arrive or positions need rebalancing, so holding times vary even when prices are observed per minute. Robots face the same pattern: often they can move with coarse updates, but when terrain becomes unstable or contacts change, they need small corrective steps to keep balance and recover. Similarly, autonomous vehicles take larger control steps on empty roads but update more frequently near intersections or in dense traffic. Across these domains, we need RL methods that learn reliably with both irregular decision times and small step sizes.

Standard reinforcement learning algorithms, however, are built on a discrete-time Markov decision process (MDP) abstraction with a fixed step size Δt. This mismatch creates practical and theoretical issues when decisions occur at irregular times and when the data mixes multiple time scales. With large Δt, the model can miss important dynamics between decisions; with very small Δt, Bellman targets change little from one step to the next, weakening learning signals and making training sensitive to noise. More fundamentally, the fixed-step Bellman recursion does not naturally account for non-uniform holding times, so the resulting updates can depend heavily on the choice of discretization rather than the underlying continuous-time control problem.

To address these issues, recent continuous-time RL work recasts learning as stochastic control. Building on the controlled-diffusion formulation of Wang et al. [25], Jia & Zhou [11] introduce the advantage-rate function $q(x, a)$, which restores state–action discrimination lost when the discrete-time Q-function collapses to V as $\Delta t \to 0$. This q-function underlies both value-based methods with martingale constraints [11] and policy-gradient variants such as entropy-regularized PPO (CPPO) [28]. These formulations provide a foundation for learning under irregular decision times and small timesteps, without relying on a fixed discretization.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

However, a central technical challenge remains. Under diffusion dynamics, minimizing the usual squared temporal-difference (TD) error does not directly enforce the dynamic programming principle: in continuous time, such losses can primarily control variance terms induced by Brownian noise rather than true Bellman errors. Prior work addresses this by imposing martingale-based objectives or martingale orthogonality conditions that characterize the optimal solution [11]. While theoretically motivated, these modifications make optimization substantially harder in practice. Martingale objectives couple q and V into a single functional optimization problem, which is difficult to simplify into standard TD-style updates and can be challenging to optimize with modern function approximation and off-policy training. This motivates seeking continuous-time learning rules that keep the stochastic-control foundation, but admit a more modular and scalable optimization structure.

In this paper, we resolve these difficulties by **decoupling** the learning of q and V, replacing a single coupled objective with a sequence of simple updates. With V fixed, the correct q is locally determined by the controlled generator. It can then be learned from trajectory data via standard regression, treating diffusion noise as estimation noise that averages out over minibatches. The second part of the decoupling is to update V from the instantaneous rate q. Because q represents an instantaneous rate rather than a finite-horizon return, the usual discrete-time backup $V(x) = \max_a Q(x, a)$ (or its soft analogue) becomes ill-posed in the continuous-time setting. To obtain the correct update, we instead interpret the continuous-time optimality equation as defining a **flow** over value functions. Discretizing this flow leads to an explicit Picard-style update driven by the entropy-regularized Hamiltonian, with a step size that indexes **algorithmic progress** rather than an environment discretization. We establish convergence of this value-flow iteration under broad conditions. Finally, although the formulation decouples q and V conceptually, the simple critic decomposition $Q \approx V + q$ allows the resulting updates to be implemented as a scalable actor-critic method, keeping the method practical while staying aligned with the continuous-time stochastic control formulation.

Contribution. Our contributions are threefold.

(1) We propose a continuous-time RL framework that separates advantage-rate learning from value-function updates, replacing martingale-coupled objectives with a simpler iterative scheme. The resulting approach integrates seamlessly with standard off-policy actor-critic training and addresses the theoretical and practical failures of discrete-time Bellman updates under irregular-time settings.

(2) We prove rigorous convergence guarantees using a probabilistic analysis technique that avoids the heavy functional-analytic machinery typically required for generator-based operators. This approach also enables us to control the approximation error directly from sampled stochastic trajectories.

(3) We provide a comprehensive empirical evaluation beyond fixed-step simulated SDE benchmarks, covering high-dimensional irregular-time control tasks and a real-world minute-level trading environment. Substantial gains over continuous-time and discrete-time baselines highlight the practical significance of our framework in irregular decision time settings.

Organization. In Section 2, we introduce the continuous-time RL formulation, together with key terminologies needed in our framework. In Section 3, we present our novel decoupled framework together with the resulting actor-critic algorithms. In Section 4, we establish theoretical and convergence guarantees for the value function, advantage-rate function, and the full algorithm. Finally, Section 5 evaluates our methods on continuous-control benchmarks and a real-world trading task under small and irregular time steps. Extended related work appears in Appendix A. Full proofs are deferred to Appendices D, E, F, G, and H. Additional experimental details are in Appendices B & J.

2 PRELIMINARIES

2.1 STOCHASTIC CONTROL FORMULATION

The usual Markov decision process (MDP) discretized with a fixed time step $\Delta > 0$ has the form:

$$X^{\Delta}_{t+\Delta} = X^{\Delta}_t + f^{\Delta}(X^{\Delta}_t, a_t), \qquad a_t \sim \pi_\alpha(\cdot \mid X^{\Delta}_t). \tag{1}$$

where $t \in \{0, \Delta, 2\Delta, \dots\}$, $X^{\Delta}_t \in \mathbb{R}^d$, and π_α denotes an exploration policy. If the increment is approximately linear in Δ, classical limit arguments [15] show that, as $\Delta \to 0$, the discrete process converges to the ODE $\dot{x}_t = f_\alpha(x_t)$, where $f_\alpha(x) := \mathbb{E}_{a\sim\pi_\alpha(\cdot|x)}[f(x, a)]$. Thus randomized actions average into the drift. This deterministic ODE limit, however, does not capture transition randomness

beyond action sampling. Adding a mean-zero perturbation ξ_t^Δ with $\mathrm{Var}(\xi_t^\Delta) = \mathcal{O}(\Delta)$ yields a SDE limit instead: accumulated perturbations converge to Brownian motion, and the limiting dynamics become a controlled diffusion (see Section I). We therefore model continuous-time RL through controlled stochastic dynamics, i.e., the state process $(X_t)_{t\geq 0} \subset \mathbb{R}^d$ evolves according to Itô SDE:

$$dX_t = b(X_t, a_t)\, dt + \sigma(X_t, a_t)\, dW_t, \qquad X_0 \sim \mu. \tag{2}$$

where W is an n-dimensional Brownian motion, $a_t \in \mathcal{A}$ is the control (action). We restrict to Markov feedback policies $a_t \sim \pi(\cdot \mid X_t)$, so exploration is implemented through randomized control.

Given discount $\beta > 0$, running reward $r : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$, and temperature $\alpha \geq 0$, we consider the entropy-regularized objective:

$$V^{(\alpha)}(x;\pi) := \mathbb{E}\left[\int_0^\infty e^{-\beta t}\Big(r(X_t, a_t) - \alpha \log \pi(a_t \mid X_t)\Big)\, dt \,\Big|\, X_0 = x\right] \tag{3}$$

with the optimal (entropy-regularized) value function $V^{(\alpha)}(x) := \sup_\pi V^{(\alpha)}(x;\pi)$.

2.2 Generators and Hamiltonians

Two standard objects from stochastic control are central to our framework: the controlled generator and the (hard/soft) Hamiltonian. For a fixed action $a \in \mathcal{A}$, define the controlled (infinitesimal) generator L^a acting on $\varphi \in C_b^2(\mathbb{R}^d)$ as:

$$(L^a\varphi)(x) := \; b(x,a) \cdot \nabla\varphi(x) + \frac{1}{2}\,\mathrm{Tr}\big(\sigma(x,a)\sigma(x,a)^\top \nabla^2\varphi(x)\big) \tag{4}$$

Using L^a, continuous-time analogue of Bellman drift, we define the action-wise Hamiltonian:

$$H_a\big(V\big)(x) := (L^a V)(x) + r(x,a) - \beta V(x) \tag{5}$$

The corresponding hard Hamiltonian ($\alpha = 0$) is:

$$H^{(0)}(V)(x) := \sup_{a\in\mathcal{A}} H_a\big(V\big)(x) \tag{6}$$

And the soft (entropy-regularized) Hamiltonian for $\alpha > 0$ is:

$$H^{(\alpha)}(V)(x) := \alpha \log \int_{\mathcal{A}} \exp\Big(\frac{1}{\alpha} H_a\big(V\big)(x)\Big)\, da \tag{7}$$

As $\alpha \to 0$, $H^{(\alpha)}$ recovers $H^{(0)}$. We include discount term $-\beta V(x)$ inside the Hamiltonian for a compact form. Under standard assumptions, optimal function $V^{(\alpha)}$ satisfies HJB equation [5; 23]:

$$H^{(\alpha)}(V^{(\alpha)})(x) = 0, \qquad x \in \mathbb{R}^d \tag{8}$$

2.3 Instantaneous q-function in continuous time

In discrete-time MDPs, the state–action value function Q provides a direct action-selection signal, e.g., through $V(x) = \max_a Q(x,a)$. In continuous time, however, the standard Q-function collapses to V as the time step shrinks to zero, i.e, $\lim_{\Delta\to 0} Q^\Delta(x,a) = V(x)$, regardless of action a. To restore action dependence, Jia & Zhou [11] introduce the *advantage rate* (instantaneous) q-function. For a policy π with value function $V^{(\alpha)}(\cdot;\pi)$, define:

$$q^{(\alpha)}(x,a;\pi) := \lim_{u\to 0} \frac{Q_u^{(\alpha)}(x,a;\pi) - V^{(\alpha)}(x;\pi)}{u} \tag{9}$$

where $Q_u^{(\alpha)}(x,a;\pi)$ is the return obtained by executing action a for a short duration u starting from state x, and then following π afterwards. With $V = V^{(\alpha)}(\cdot;\pi)$, standard small-time expansion yields:

$$Q_u^{(\alpha)}(x,a;\pi) = V(x) + H_a(V)(x)\, u + o(u) \tag{10}$$

As a result, q-function can be also expressed as:

$$q_V(x,a) = q^{(\alpha)}(x,a;\pi) = H_a(V)(x) \tag{11}$$

Unlike the degenerate continuous-time limit of Q, the function $q(x, a)$ retains action dependence and provides the correct local improvement signal.

Notations. From here on, we write $V^{(\alpha)}$ for the **optimal** value function, and use V for a generic value function (e.g., under a non-optimal policy or an approximation) in place of $V^{(\alpha)}(x; \pi)$. By Equation (11), q-function is determined by the corresponding V, so we write $q = q_V$ instead of $q^{(\alpha)}(x, a; \pi)$ so that $q_V(x, a) = H_a(V)(x)$. In particular, the optimal q-function is denoted as $q_{V^{(\alpha)}}$.

3 FORMULATION

3.1 CHALLENGES IN CONTINUOUS-TIME RL

The instantaneous q-function introduced in Section 2.3 resolves the degeneracy of the discrete-time Q-function in the infinitesimal limit. A second obstacle remains. Under diffusion dynamics, the short-time value increment contains both a Bellman drift term and a stochastic martingale fluctuation induced by Brownian noise. Over a holding interval u, the Bellman drift scales as $O(u)$, while the stochastic increment scales as $O(\sqrt{u})$. Consequently, a squared short-time TD residual is dominated by martingale variation rather than Bellman inconsistency so that naively applying discrete-time TD learning no longer aligns with the continuous-time dynamic programming principle.

Prior work [11] addresses this issue by characterizing optimality through martingales, enforced through long-horizon martingale losses or orthogonality tests against adapted test processes. While theoretically principled, these approaches couple V and q into a single functional optimization problem, depend on the choice of test processes, and are difficult to reduce to standard off-policy actor-critic updates. Continuous-time policy-gradient methods such as CPPO inherit the same martingale-enforcement structure. This creates a common bottleneck for existing continuous-time RL methods, reflected empirically by their weaker performance even in irregular-time settings.

Our formulation avoids this bottleneck by separating the roles of q and V. With V fixed, we estimate q from conditional short-time value increments; the martingale term has zero conditional mean and becomes minibatch estimation noise rather than a pathwise optimization target. We then update V from this instantaneous rate through a Hamiltonian flow, instead of minimizing a noisy TD residual or applying a degenerate discrete-time backup. This preserves the stochastic-control foundation while replacing coupled martingale optimization with sequential rate estimation and value-flow updates.

3.2 CONTINUOUS-TIME ACTOR-CRITIC VIA HAMILTONIAN FLOW

Instead of enforcing martingale constraints through a single coupled optimization over (V, q), we break learning into a sequence of smaller, tractable subproblems that iteratively update q and V:

$$V_0 \;\Rightarrow\; q_0 \;\Rightarrow\; V_1 \;\Rightarrow\; q_1 \;\Rightarrow\; \cdots \tag{12}$$

The guiding principle is simple: *learning q is easy if V is fixed*, and the remaining difficulty is *updating V from an instantaneous rate* without degeneracy.

Step 1: Updating q from a fixed V. For a fixed value function V, from Equation (11), the q-function can be determined locally by the controlled diffusion generator L^a:

$$q_V(x, a) = r(x, a) + (L^a V)(x) - \beta V(x). \tag{13}$$

The characterization is exact, but not model-free since $(L^a V)(x)$ depends on unknown dynamics.

Step 2: Updating V from q is not a discrete-time max step. In discrete-time RL, one typically recovers V via a hard or soft maximization of Q:

$$\begin{aligned} V(x) &= \max_a Q(x, a), \text{ or} \\ V(x) &= \alpha \log \int_{\mathcal{A}} \exp\big(Q(x, a)/\alpha\big)\, da \end{aligned} \tag{14}$$

In continuous time, however, q is an instantaneous *rate*, not a value. If we relate a small-step return by $Q_u(x,a) \approx V(x) + u\,q(x,a)$ as $u \to 0$, then the soft backup degenerates:

$$\begin{aligned} V(x) &= \alpha \log \int_{\mathcal{A}} \exp\big((V(x) + u\,q(x,a))/\alpha\big)\,da \\ &= V(x) + \alpha \log \int_{\mathcal{A}} \exp\big(u\,q(x,a)/\alpha\big)\,da \end{aligned} \tag{15}$$

As $u \to 0$, this becomes a trivial update $V_{\text{new}}(x) \to V(x)$, while enforcing equality yields a meaningless normalization condition. This reflects a mismatch: V behaves like a displacement, while q behaves like a velocity, so direct max/softmax backups are ill-posed in the infinitesimal setting.

To extract a non-degenerate update from q in the infinitesimal setting, we propose a novel approach that interprets the update not as a discrete backup but as a *flow* indexed by an iteration parameter τ. As a first step, consider the soft backup increment at small u:

$$V_u(x) := V(x) + \alpha \log \int_{\mathcal{A}} \exp\big(u\,q(x,a)/\alpha\big)\,da \tag{16}$$

The increment vanishes as $u \to 0$. However, Jensen inequality implies an overestimation for V_u:

$$V_u(x) - V(x) \;\le\; \alpha u \,\log \int_{\mathcal{A}} \exp(q(x,a)/\alpha)\;da \tag{17}$$

Crucially, this overestimation allows us to derive an update gradient flow by considering the ratio $[V_u(x) - V(x)]/u$:

$$\frac{d}{d\tau} V_\tau(x) = \lim_{u \to 0} [V_u(x) - V(x)]/u \approx \mathcal{Q}^{(\alpha)}(q_{V_\tau})(x). \tag{18}$$

where the aggregation operator $\mathcal{Q}$ acts on q as:

$$\begin{aligned} \mathcal{Q}^{(\alpha)}(q)(x) &:= \alpha \log \int_{\mathcal{A}} \exp\big(q(x,a)/\alpha\big)\,da, \\ \mathcal{Q}^{(0)}(q)(x) &:= \sup_{a \in \mathcal{A}} q(x,a). \end{aligned} \tag{19}$$

Under our decoupled scheme, $q_\tau = q_{V_\tau}$ is defined solely by V_τ, and substituting this into Equation (19) yields exactly the Hamiltonian operator:

$$\mathcal{Q}^{(\alpha)}(q_V)(x) = H^{(\alpha)}(V)(x). \tag{20}$$

Therefore the flow becomes the Hamiltonian-driven ODE:

$$\frac{d}{d\tau} V_\tau(x) = H^{(\alpha)}(V_\tau)(x), \tag{21}$$

Discretizing this gives us a **Picard–Hamiltonian** iteration with step size $\tau > 0$:

$$V^{\text{new}}(x) = V(x) + \tau\, H^{(\alpha)}(V)(x) = T_\tau^{(\alpha)}(V)(x). \tag{22}$$

In Section 4, we establish convergence of this *ideal* iteration to $V^{(\alpha)}$ as $\tau \to 0$.

Model-free q-estimation induces discretized Hamiltonian flows. The remaining gap is that $(L^a V)(x)$ in Equation (13) is not available in model-free RL. Instead, we estimate the instantaneous rate via its small-time representation: for a holding time $u > 0$,

$$\begin{aligned} q_V(x,a) &= \lim_{u \to 0} q_V^u(x,a), \\ q_V^u(x,a) &:= \frac{e^{-\beta u}\,\mathbb{E}[V(X_{t+u}) \mid X_t = x, a_t = a] - V(x)}{u} + r(x,a). \end{aligned} \tag{23}$$

This replaces the generator by a short-horizon value increment and converges to q_V as $u \to 0$. To reduce discretization bias, we further apply Richardson extrapolation:

$$\tilde{q}_V^u(x,a) := 2\,q_V^{u/2}(x,a) - q_V^u(x,a). \tag{24}$$

These approximations induce "discretized" Hamiltonians:

$$H_u^{(\alpha)}(V) := \mathcal{Q}^{(\alpha)}(q_V^u), \qquad \tilde{H}_u^{(\alpha)}(V) := \mathcal{Q}^{(\alpha)}(\tilde{q}_V^u), \tag{25}$$

and the corresponding value iterations

$$V_{k+1} = V_k + \tau\, H_u^{(\alpha)}(V_k), \qquad V_{k+1} = V_k + \tau\, \tilde{H}_u^{(\alpha)}(V_k). \tag{26}$$

In Section 4, we prove convergence of these discretized flows (including the Richardson variant) and show that $\tilde{q}_{V_k}^u$ converges to the optimal $q_{V^{(\alpha)}}$ as $(\tau, u) \to 0$.

Algorithm 1 Continuous-time soft actor-critic (CT-SAC)

1: **Input:** discount β, temperature α, Euler step $\tau > 0$.
2: Initialize parameters θ and ϕ for Q_θ and policy π_ϕ.
3: **for** $k = 0, 1, 2, \dots$ **do**
4: Collect transitions (x, a, r, x', u), where $x = X_t$, u is the time step increment, $x' = X_{t+u}$, $a = a_t$ is the action, and $r = r(X_t, a_t)$ from rollouts under current policy π_ϕ and store in an off-policy buffer. u can vary across samples.
5: **Update** $Q_\theta = q_\theta + V_\theta$**:** Fit Q_θ using mini-batch samples (x, a, r, x', u) with targets being the right hand-side of Equation (28).
6: **Update** π_ϕ: towards $\pi_\phi(a \mid x) \propto \exp\big(Q_\theta(x, a)/\alpha\big)$
7: **end for**

3.3 Final algorithm: a single critic via $Q \approx V + q$

The decoupled iteration above is conceptually clean: learn q given V, then update V via the Hamiltonian flow. However, maintaining two separate function approximators is inconvenient in practice. We therefore represent both quantities with a single critic network:

$$Q_k(x, a) := V_k(x) + q_k(x, a). \tag{27}$$

This is a numerical parameterization: while V and q have different roles, the decomposition lets one network carry both state-value and advantage-rate information. Near optimality, $q_k(x, a)$ is typically small for actions drawn from the improved policy, so $Q_k(x, a) \approx V_k(x)$ on-policy, making a single critic sufficient. With this representation, the decoupled update yields a practical sample-based critic target. By defining $\tilde{Q}_k(x, a) := Q_k(x, a) - \alpha \log \pi(a \mid x)$, the critic update takes the form:

$$\begin{aligned} Q_{k+1}(x, a) = (1 - \tau) Q_k(x, a) + \tau \mathbb{E}_a\big[\tilde{Q}_k(x, a)\big] \\ + \tau r(x, a) + \tau \frac{\gamma\, \mathbb{E}_{a'}\big[\tilde{Q}_k(x', a')\big] - \mathbb{E}_a\big[\tilde{Q}_k(x, a)\big]}{u} \end{aligned} \tag{28}$$

where discount $\gamma = e^{-\beta u}$, and u is the (possibly irregular) transition duration in the sampled tuple (x, a, r, x', u). By variational form of the soft Hamiltonian (Appendix I), the maximizer over $\pi(\cdot \mid x)$ is a Boltzmann policy, $\pi_V^{(\alpha)}(a \mid x) \propto \exp\big(q_V(x, a)/\alpha\big)$. Since $Q = V + q$ differs from q only by an a-independent shift, we implement the actor update equivalently as $\pi(a \mid x) \propto \exp\big(Q(x, a)/\alpha\big)$.

Combining these gives a continuous-time soft actor-critic method (CT-SAC; Algorithm 1). A deterministic variant based on hard operator gives a continuous analogue of TD3 (CT-TD3; Algorithm 2).

4 Theoretical analysis

Our theoretical analysis for framework in Section 3 proceeds in three layers. We first study the *ideal* Picard–Hamiltonian iteration $V_{k+1} = T_\tau^{(\alpha)}(V_k) = V_k + \tau H^{(\alpha)}(V_k)$, which corresponds to the analytic rate q_V. We then analyze the *model-free discretized flows* obtained by replacing q_V with its finite-horizon approximation q_V^u, and its Richardson-corrected version $\tilde{q}_V^u$. Finally, we lift these results to the single-critic implementation of CT-SAC/CT-TD3.

Because of the infinitesimal generator term $L^a V$, unlike discrete-time Bellman operators, $T_\tau^{(\alpha)}$ is not a **contraction** mapping on $(C_b, \|\cdot\|_\infty)$ and purely analytic approaches often require heavy machinery. Instead, we compare $T_\tau^{(\alpha)}$ to a probabilistic dynamic-programming operator that is a sup-norm contraction and treat $T_\tau^{(\alpha)}$ as a controlled local approximation. We define the entropy-regularized dynamic-programming semigroup $\{\Phi_\tau^{(\alpha)}\}_{\tau \geq 0}$:

$$(\Phi_\tau^{(\alpha)} V)(x) := \sup_\pi \mathbb{E}\left[\int_0^\tau e^{-\beta t}\Big(r(X_t, a_t) - \alpha \log \pi(a_t \mid X_t)\Big)\, dt + e^{-\beta\tau} V(X_\tau) \,\Big|\, X_0 = x\right] \tag{29}$$

The semigroup $\Phi_\tau^{(\alpha)}$ is value-only and serves as contraction reference; it does not provide action-dependent signal needed for actor-critic updates. We begin with the ideal iteration's convergence.

Theorem 4.1 (Value Function Convergence). *Fix $\alpha \geq 0$ and $\tau > 0$. Under Assumptions C.1, C.2, and C.3, let V_k be the value function after k iterations $V_{k+1} := T_\tau^{(\alpha)}(V_k)$. Then, V_k converges to $V^{(\alpha)}$ as $\tau \to 0$, and for all $k \geq 0$,*

$$\|V_k - V^{(\alpha)}\|_\infty \leq C_0 e^{-\beta\tau k} + \frac{C_1 \tau^{3/2}}{1 - e^{-\beta\tau}}. \tag{30}$$

Here C_0 and C_1 are constants independent of k.

Proof sketch. We compare the Picard–Hamiltonian update $V_{k+1} = T_\tau^{(\alpha)}(V_k)$ with the dynamic-programming semigroup iteration $W_{k+1} = \Phi_\tau^{(\alpha)}(W_k)$. A short-time Itô expansion gives the local operator error $\left\|\Phi_\tau^{(\alpha)}(V) - T_\tau^{(\alpha)}(V)\right\|_\infty \leq C_\alpha \tau^{3/2}$. Since $\Phi_\tau^{(\alpha)}$ is a contraction, this local error accumulates geometrically, yielding $\|V_k - W_k\|_\infty \leq C_\alpha \tau^{3/2}/(1 - e^{-\beta\tau})$. Finally, W_k converges exponentially to $V^{(\alpha)}$, and the triangle inequality gives the stated bound.

We extend the result to the model-free estimator q_V^u (and $\tilde{q}_V^u$). We state the theorem for fixed u, and defer the random-increment case U to the Appendix G.

Theorem 4.2 (Convergence with q-function). *Under Assumptions E.2 and E.3, as $k \to \infty$, the Richardson iteration $V_{k+1} = V_k + \tau \tilde{H}_u^{(\alpha)}(V_k)$ satisfies $\|V_k - V^{(\alpha)}\|_\infty = \mathcal{O}(\tau + u^2)$ and $\|\tilde{q}_{V_k}^u - q_{V^{(\alpha)}}\|_\infty = \mathcal{O}(\tau/u + u)$. In addition, the first-order iteration $V_{k+1} = V_k + \tau H_u^{(\alpha)}(V_k)$ satisfies $\|V_k - V^{(\alpha)}\|_\infty = \mathcal{O}(\tau + u)$ and $\|q_{V_k}^u - q_{V^{(\alpha)}}\|_\infty = \mathcal{O}\big(u^{-1/2}(\tau + u)^{3/4} + \tau + u\big)$.*

In particular, if $\tau, u \to 0$ and $\tau/u \to 0$, then both the value estimation V_k and q-function estimates converge to the optimal values $V^{(\alpha)}$ and $q_{V^{(\alpha)}}$.

Proof sketch. Similar to Theorem 4.1, by comparing both model-free updates with the contraction semigroup $\Phi_\tau^{(\alpha)}$, the Richardson and first-order one-step errors $\mathcal{O}(\tau^2 + \tau u^2)$ and $\mathcal{O}(\tau^2 + \tau u)$ yield the stated value bounds. For Richardson case, the $\mathcal{O}(1/u)$ stability bound together with the $\mathcal{O}(u^2)$ modelling bias gives the desired rate $\mathcal{O}(\tau/u + u)$ for q-function. For the first-order estimator, through the interpolation inequality, we get the stability of order $u^{-1/2}\|V_k - V^{(\alpha)}\|_\infty^{3/4}$. Together with the $\mathcal{O}(u)$ modelling bias, this gives the stated first-order q-bound for this case.

Finally, we translate these guarantees to single-critic $Q_k = V_k + q_k$ underlying CT-SAC/CT-TD3.

Theorem 4.3 (Theoretical Algorithm Convergence). *Consider the exact single-critic iteration Q_k, for either Equation* (28) *or its Richardson-corrected variant, with no optimization or statistical error. Then as $k \to \infty$, Q_k converges to the optimal value $Q^{(\alpha)} = V^{(\alpha)} + q_{V^{(\alpha)}}$.*

Corollary 4.4 (Algorithm Convergence). *Suppose at each iteration k, n_k samples are used in the gradient update. Then, for any pair of $(\epsilon, \delta) > 0$, there exists an iteration L together with the sequence $n_1, n_2, \cdots, n_L$ so that with the probability of at least $1 - \delta$, $\mathbb{E}\left[|Q_L(X, A) - Q^{(\alpha)}(X, A)|^2\right] < \epsilon$ for random state-action pair (X, A). Here $Q^{(\alpha)} = V^{(\alpha)} + q_{V^{(\alpha)}}$ is the optimal function, and Q_L is the learned function from Algorithm 1.*

Proof sketch. We construct the corresponding sequences (V_k, q_k) induced by the exact update and prove by induction that the decomposition (27) is preserved, i.e., $Q_k = V_k + q_k$ for all k. The convergence then follows by Theorem 4.2. For statistical part, through the Lipschitzness of update mapping $\mathcal{F}$, we can derive the inductive inequality through the statistical error: $\epsilon_{k+1} \leq \mathrm{stat}_k + \left(1 + \frac{\tau}{u}\right)\epsilon_k$ for $\epsilon_k = \left\|\hat{Q}_k - Q_k\right\|$ with statistical regression error stat_k. For large enough number of samples, $\mathrm{stat}_k \to 0$, yielding the convergence of ϵ_k, and, therefore, of the error for $Q^{(\alpha)}$.

Refer to Appendices D, E, and F for the full detailed proofs of Theorems 4.1, 4.2, and 4.3 respectively.

5 Experiments

Moving beyond fixed-step simulated SDE benchmarks, we evaluate continuous-time RL under irregular decision times on standard control benchmarks and a real-world trading task.

5.1 TASKS AND BASELINES

Irregular-time setup. Across all tasks, the agent acts at event times $t_0 < t_1 < \cdots$ with holding times $u_k := t_{k+1} - t_k$, where u_k **mixes between small and large timesteps** within the same rollout. Concretely, we implement this by sampling u_k around the nominal control step u_{nom}. For example, in `Cheetah`, holding times can be as small as $u_{\text{nom}}/5$ and as large as $3u_{\text{nom}}$. This is harder than fixed-step training as trajectories are not time-aligned and learning must generalize across step sizes.

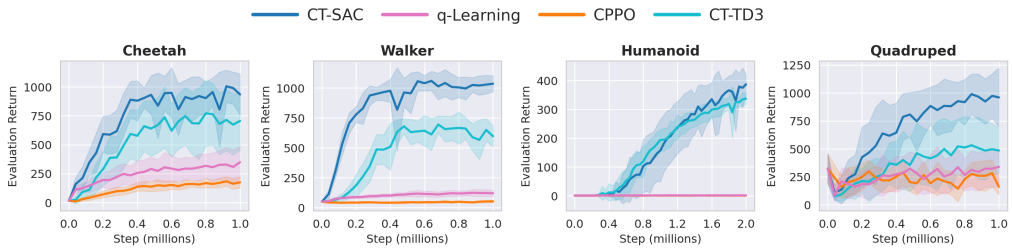


Figure 1: Evaluation returns for 4 continuous-time algorithms on control tasks over 12 seeds.

Control tasks. We use DeepMind Control Suite [24] for high-dimensional continuous control. Irregular timing is particularly important for humanoid and quadruped locomotion, where stability relies on timely corrective actions.

Real-world trading. We additionally evaluate on a trading environment constructed from minute-resolution market data (Alpaca API). The universe contains 20 large-cap tickers across four sectors, and each episode spans two trading weeks. The agent trades at irregular intervals with holding times $u \in [1, 20]$ minutes, using multi-scale price features. The action is to buy/sell a bounded quantity within an industry group, with position limits and transaction costs. The reward is the realized profit over the subsequent holding interval, so episode return equals total two-week P&L. To avoid temporal leakage, we train on Q3 2023–Q2 2025 and evaluate on Q3 2025 using a fixed set of episodes so all methods face the same market paths. Details are provided in Appendix B.

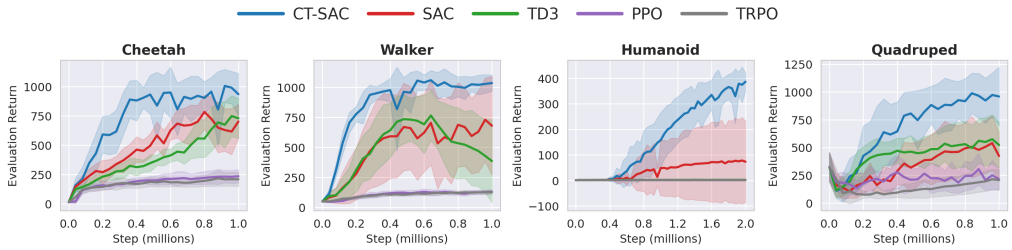


Figure 2: Evaluation returns for our CT-SAC against 4 discrete-time algorithms on control tasks.

Baselines. We benchmark against two baseline groups: continuous-time and discrete-time. For continuous-time RL, we include representative advantage-rate methods with martingale enforcement (q-Learning) [11], as well as a continuous-time policy-gradient baseline (CPPO) [28]. To our knowledge, most existing continuous-time deep RL algorithms fall into these two families. Because these methods can be optimization-sensitive, we additionally apply stabilization techniques such as gradient clipping to improve their performance. For discrete-time RL, we compare against widely used state-of-the-art algorithms, SAC, TD3, TRPO, and PPO, implemented in Stable-Baselines3 [16].

Evaluation. We train each method for 1M environment steps on control tasks (2M on `humanoid`). We tune hyperparameters over the search space and select the best configuration for each method. We additionally keep the top 3 configurations for ablations, and report the best one in the main figures. Evaluation is performed periodically using 10 episodes per checkpoint for control tasks and 30 episodes for trading. We report mean learning curves and aggregate metrics over 12 random seeds.

Runtime & complexity. Details are given in Appendix B.

5.2 RESULTS

Continuous-time baselines. Figure 1 compares our CT-SAC and CT-TD3 with prior continuous-time baselines across the control suite. Our methods achieve the strongest overall performance and more stable learning. These gains are consistent with our decoupled design that decomposes learning into simpler subproblems and avoids the optimization burden of coupled martingale objectives.

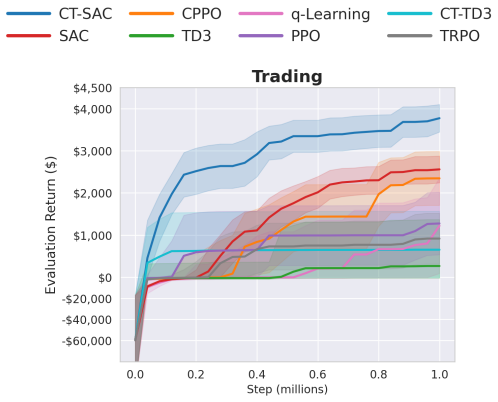


Figure 3: 2-week evaluation returns for CT-SAC and CT-TD3 against 6 other algorithms on trading.

Discrete-time baselines. CT-SAC also outperforms leading discrete-time algorithms (Figures 2 and 3). When step sizes vary across samples and between training and evaluation, discrete-time updates degrade, while our continuous-time update remains robust. Across control and trading, CT-SAC improves earlier and stabilizes faster than discrete-time baselines. CT-TD3 trails CT-SAC but outperforms TD3, suggesting that both continuous-time updates and entropy regularization matter.

Performance on trading. On control tasks, CPPO is weaker than discrete-time baselines, but on trading it becomes competitive with strong off-policy methods such as SAC. This suggests that continuous-time modeling can matter more in trading under irregular decision times. CT-SAC remains the best method overall: using only price-derived features (no news or fundamentals data), it achieves about $3{,}500$ average profit per two-week episode from a $100{,}000$ initial portfolio (Figure 3). Over Q3 2025, it achieves roughly $21{,}000$ total profit, nearly doubling the second-best method.

Finally, statistical tests over 12 seeds confirm statistical significant improvements across tasks.

Ablation study. We run ablations using the top 3 tuned hyperparameter configurations for each method. CT-SAC remains the top performer even outside its best configuration.

Duration-aware SAC baselines. We further compare against 2 irregular-time adaptations of SAC that account for elapsed time through reward scaling and duration-aware discounting. These variants perform similarly to standard counterparts and consistently rank behind CT-SAC (Figure 4 and 5).

6 CONCLUSION

We introduce a continuous-time reinforcement learning framework for irregular decision times. The core idea is a simplified learning procedure: estimate an instantaneous advantage-rate signal from rollouts under a fixed value predictor, then update the value predictor via a Hamiltonian flow. This avoids the coupled martingale objectives used in prior continuous-time methods. A single-critic implementation makes the approach compatible with standard off-policy actor–critic training and stable under variable step sizes. On the theory side, we establish convergence guarantees despite the presence of infinitesimal generators, where discrete-time contraction arguments no longer apply. Our analysis uses a probabilistic technique that links the flow update to the dynamic-programming semigroup and extends naturally to the model-free setting with finite time increments. Empirically, we evaluate our method under an irregular-time setup with many short steps mixed with large jumps, on high-dimensional control benchmarks and a real-world trading task. Across these settings, our method consistently outperforms existing continuous-time baselines and strong discrete-time algorithms.

AI USE STATEMENT

Generative AI tools were used only to aid and polish the writing of this paper.

ETHICS STATEMENT

This work studies reinforcement-learning methods for continuous-time and irregular-time decision making. The work does not involve human subjects or interventions on individuals.

REPRODUCIBILITY STATEMENT

We provide the complete code at `https://anonymous.4open.science/r/ct-rl`. Appendices B and J provide implementation details, experimental settings, hyperparameters, and additional results. Complete derivations and proofs for the theoretical results are provided in the corresponding theoretical appendices.

REFERENCES

[1] Samuel Ainsworth, Kendall Lowrey, John Thickstun, Zaid Harchaoui, and Siddhartha Srinivasa. Faster policy learning with continuous-time gradients. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pp. 1054–1067. PMLR, 07 – 08 June 2021.

[2] Alpaca Markets. Alpaca market data api documentation. Available at `https://docs.alpaca.markets/docs/market-data-api`, 2026. Accessed: 2026-01-27.

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

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

[5] Wendell H Fleming and H Mete Soner. *Controlled Markov processes and viscosity solutions*. Springer, New York, NY, 2nd edition, 2006.

[6] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018.

[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870, Vienna, Austria, 10–15 Jul 2018. PMLR.

[8] Chengxiu Hua, Jiawen Gu, and Yushun Tang. Continuous q-score matching: Diffusion guided reinforcement learning for continuous-time control. *arXiv preprint arXiv:2510.17122*, 2025.

[9] 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, 2022.

[10] 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, 2022.

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

[12] Jaeyoung Lee and Richard S. Sutton. Policy iterations for reinforcement learning problems in continuous time and space — fundamental theory and methods. *Automatica*, 126:109421, 2021. ISSN 0005-1098.

[13] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016.

[14] D. S. Mitrinović, J. E. Pečarić, and A. M. Fink. *Inequalities Involving Functions and Their Integrals and Derivatives*, volume 53 of *Mathematics and Its Applications*. Springer, Dordrecht, 1991.

[15] Rémi Munos. Policy gradient in continuous time. *Journal of Machine Learning Research*, 7 (27):771–791, 2006.

[16] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[17] Patrick Rebeschini. Algorithmic foundations of learning, 2022. URL `https://www.stats.ox.ac.uk/%7Erebeschi/teaching/AFoL/22/`.

[18] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[20] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. Bradford Books, Cambridge, MA, 2 edition, 2018.

[21] Corentin Tallec, Léonard Blier, and Yann Ollivier. Making deep Q-learning methods robust to time discretization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6096–6104. PMLR, 09–15 Jun 2019.

[22] Wenpin Tang and Xun Yu Zhou. Regret of exploratory policy improvement and q-learning. *arXiv preprint arXiv:2411.01302*, 2024.

[23] Wenpin Tang, Paul Yuming Zhang, and Xun Yu Zhou. Exploratory HJB equations and their convergence. *arXiv preprint arXiv:2109.10269*, 2021.

[24] Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638.

[25] 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.

[26] Harley Wiltzer, Marc G. Bellemare, David Meger, Patrick Shafto, and Yash Jhaveri. Action gaps and advantages in continuous-time distributional reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47815–47848. Curran Associates, Inc., 2024.

[27] Cagatay Yildiz, Markus Heinonen, and Harri Lähdesmäki. Continuous-time model-based reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12009–12018. PMLR, 18–24 Jul 2021.

[28] Hanyang Zhao, Wenpin Tang, and David Yao. Policy optimization for continuous reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 13637–13663. Curran Associates, Inc., 2023.

[29] Hanyang Zhao, Haoxian Chen, Ji Zhang, David Yao, and Wenpin Tang. Score as action: Fine tuning diffusion generative models by continuous-time reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 77371–77389. PMLR, 13–19 Jul 2025.

A Related works

Discrete-time reinforcement learning. Most effective deep reinforcement learning for continuous control can be broadly grouped into (i) off-policy actor–critic methods and (ii) on-policy policy-gradient methods [20], with widely used representatives including Soft Actor–Critic (SAC) [7], Twin Delayed Deep Deterministic Policy Gradient (TD3) [6], Trust Region Policy Optimization (TRPO) [18], and Proximal Policy Optimization (PPO) [19], while many other practical algorithms can be viewed as variants, hybrids, or refinements of these cornerstones. SAC is particularly powerful due to its entropy-regularized objective, which encourages exploration while yielding stable policy improvement and strong empirical performance in high-dimensional continuous-action domains. In contrast to SAC's stochastic policy, TD3 is a deterministic actor–critic method that can be viewed as a strengthened successor of DDPG [13], mitigating overestimation bias via clipped double critics and delayed policy updates. On the policy-gradient side, TRPO is notable for its principled trust-region formulation and theoretical guarantees, whereas PPO is widely adopted as an efficient and scalable approximation that performs robustly across a wide range of tasks. Despite their success, all of these methods are fundamentally formulated on fixed-step Markov decision processes (MDPs). When the decision interval Δt becomes small or irregular, the Bellman recursion becomes increasingly sensitive to discretization [21; 11]. Additionally, learning signals can degenerate in the settings mentioned in Section 1, motivating new approaches that are more suitable for continuous-time problems.

Early continuous-time RL and ODE-based approaches. Continuous-time reinforcement learning has long been studied from a control-theoretic perspective, primarily through the Hamilton–Jacobi–Bellman (HJB) equation [5]. Early work by Baird [3], Doya [4], and Munos [15] explored continuous-time value learning, policy gradients, and advantage-based updates as alternatives to fixed-step MDP formulations. However, these early approaches are typically restricted to deterministic dynamical systems or require additional structure beyond standard model-free RL. For instance, they need access to the closed-form transition dynamics or their derivatives to construct path-wise or value-gradient updates. Subsequent developments [12; 1; 27] further refine continuous-time analogues of actor–critic or policy-gradient methods through closed-form formulae. Nonetheless, these methods also inherit similar limitations, remaining either model-based, tied to deterministic dynamics, or relying on assumptions that are difficult to meet in general stochastic environments under irregular decision times. A central limitation is that the continuous-time problem demands a stochastic-control formulation, which should be associated with stochastic differential equations (SDEs) rather than deterministic ODEs, as discussed in Section 2.1. Without a framework that works well under SDEs, naïve temporal-difference (TD) learning can lead to minimizing the variance terms induced by Brownian noise rather than true dynamic-programming errors, a phenomenon highlighted in [9; 11] and analyzed further in Section 3.1. Moreover, as the timestep shrinks, the discrete-time Q-function collapses to the value function, eliminating meaningful action ranking. This collapse is empirically suggested by Tallec et al. [21], who demonstrate the sensitivity of Q-learning-based methods to time discretization. Nevertheless, that work still operates in a discrete setting with fixed time step, rather than a continuous-time setup under stochastic dynamics.

Stochastic-control formulations and recent continuous-time RL. A more recent novel line of work [25; 9; 10; 11; 28] formulates reinforcement learning directly as stochastic control problem under diffusion dynamics. Wang et al. [25] introduce an entropy-regularized continuous-time framework that models exploration through relaxed controls and establishes a stochastic HJB characterization. Tang et al. [23] further prove important convergence results for the associated stochastic HJB. Building on this foundation, Jia & Zhou [11] show that the conventional state–action value function does not survive the continuous-time limit and propose the advantage-rate (or "little-q") function as the correct infinitesimal analogue. This quantity is closely linked to the Hamiltonian and admits characterization via martingale conditions, enabling continuous-time analogues of policy evaluation and improvement. Crucially, such martingale conditions arise precisely to address the failure modes of naïve TD updates under diffusion noise, as discussed in Section 3.1. Within the same stochastic-control framework, [28] further develops policy optimization methods, deriving continuous-time counterparts of TRPO and PPO based on occupation measures and performance-difference formulas. Complementary theoretical work, including regret analyses [22] and the study of distributional quantities such as the action-gap [26], further enriches this emerging line of work and clarifies the theoretical behavior of continuous-time learning. In terms of new algorithmic contributions, recent works include [8], which follows

the martingale-based characterization in [11], and [29], which extends policy-gradient methods [28] using diffusion-based parameterizations. These works, however, specialize the policy class through diffusion modeling or apply similar martingale objectives to other tasks, rather than proposing a fundamentally new learning approach to address the complexity of martingale characterization. In contrast, our work focuses on developing a new decoupled learning scheme that preserves the continuous-time theoretical foundation while yielding a more practical optimization structure. As shown in Section 4 and Section 5, this decoupling leads to an efficient algorithm with theoretical guarantees and strong empirical performance on advanced control benchmarks and a real-world trading task under small and irregular time steps.

B EXPERIMENT DETAILS

B.1 TASK DETAILS

B.1.1 TRADING

Market data and episode construction. We consider a diversified equity universe consisting of 4 major industry sectors, each containing 5 liquid, large-cap tickers (20 total). We use minute-resolution OHLCV bars and define one episode as two weeks of trading, i.e., 390 minutes per day over 5 trading days per week. Decisions are made at *irregular* times: after each action, the environment samples a holding time $u \in [1, 20]$ minutes and advances the market by u minutes before the next decision.

State, action, and reward. The state summarizes recent market behavior through multi-scale historical features (e.g., rolling windows spanning minutes, hours, and days) computed from the minute bars. At each decision time, the agent chooses a bounded buy/sell quantity within a sector-specific group, subject to position limits and proportional transaction costs. Rewards are defined as realized profit over the subsequent holding interval; consequently, the episode return equals the total P&L accumulated over the two-week horizon.

Train–test split. To avoid temporal leakage, we train on a contiguous historical window from Q3 2023 through Q2 2025 and evaluate on a disjoint quarter (Q3 2025). Evaluation uses a *fixed set* of episodes spanning the full quarter, so all methods are tested on the same market-path distribution and performance differences cannot be attributed to favorable evaluation windows. Minute-level data are obtained from the Alpaca API [2]. For reproducibility, we provide preprocessed datasets (`train.npz` and `eval.npz`) under `data/trading/processed_data/` along with scripts and instructions to regenerate the minute-resolution features from raw data.

B.1.2 CONTROL TASKS

We evaluate on four standard continuous-control benchmarks from the DeepMind Control Suite (DMC) [24]: `cheetah-run`, `walker-run`, `humanoid-walk`, and `quadruped-run`. DMC provides a physics-based simulator with direct control over integration and control rates, which makes it well-suited for our irregular-time setting. In DMC, rewards are normalized to $[0, 1]$; thus the maximum achievable return is approximately proportional to the episode duration.

B.1.3 IRREGULAR-TIME CONFIGURATION

A central goal of this work is to evaluate learning under *intertwined* time scales, where trajectories contain a mixture of very small and very large time steps within the same episode. For each task, we report the finest time scale $\Delta t_{\text{physics}}$, the minimum and maximum decision increments $(\Delta t_{\min}, \Delta t_{\max})$, and the maximum episode duration $T_{\max}$, along with the fractions of *small*, *large*, and *average* steps and the range of number of decision steps per episode. Control-task time is measured in seconds, while trading time is measured in minutes (See Table 1).

B.2 ABLATION STUDY TABLES

We report hyperparameter ablations in Table 2. Across all environments, **CT-SAC is consistently the best-performing method among the tested configurations**, and the same trend holds for **CT-TD3 vs. TD3**. To keep the main ablation table compact, we show the **top/second/third** configurations per

Table 1: Irregular-time sampling configuration per task. We include finest time scale $\Delta t_{\text{physics}}$, decision-step bounds $(\Delta t_{\min}, \Delta t_{\max})$, maximum episode duration $T_{\max}$, step-type proportions (small/large/average), and the step range per episode.

Task	$\Delta t_{\text{physics}}$	$\Delta t_{\min}$	$\Delta t_{\max}$	% small	% large	% avg	$T_{\max}$	# of steps range
Cheetah	0.0020	0.002	0.030	89.1%	9.9%	1.0%	10	1200-2000
Walker	0.0025	0.005	0.075	89.1%	9.9%	1.0%	25	1200-2000
Humanoid	0.0050	0.010	0.040	40.0%	40.0%	20.0%	25	800-1000
Quadruped	0.0050	0.005	0.050	89.1%	9.9%	1.0%	20	1200-2000
Trading	1	1	11	89.1%	9.9%	1.0%	4000	1600-2100

algorithm and task; the exact hyperparameter settings corresponding to these entries are provided in the hyperparameter dictionary in Section J.

Table 2: Hyperparameter ablations across all tasks. For each algorithm and task, we report the returns of the **top**, **second**, and **third** configurations from the hyperparameter tuning process.

(a) CT-SAC versus discrete-time algorithms SAC and TD3

	CT-SAC			SAC			TD3		
Task	**top**	**second**	**third**	**top**	**second**	**third**	**top**	**second**	**third**
Cheetah	**934.76**	**863.45**	**807.19**	701.88	680.30	605.63	730.12	694.15	692.81
Walker	**1035.52**	**928.26**	**817.28**	680.08	560.98	522.85	385.73	286.09	191.74
Humanoid	**386.75**	**379.75**	**371.75**	73.39	39.28	2.12	2.28	2.24	2.11
Quadruped	**959.75**	**958.44**	**829.41**	423.33	314.77	284.00	522.63	518.37	464.95
Trading	**37.72**	**33.09**	**31.98**	25.59	21.04	13.77	2.67	-5.46	-9.46

(b) CT-SAC versus discrete-time algorithms TRPO and PPO

	CT-SAC			TRPO			PPO		
Task	**top**	**second**	**third**	**top**	**second**	**third**	**top**	**second**	**third**
Cheetah	**934.76**	**863.45**	**807.19**	210.10	190.51	170.73	235.61	208.42	206.70
Walker	**1035.52**	**928.26**	**817.28**	126.73	119.95	114.90	129.94	120.28	118.08
Humanoid	**386.75**	**379.75**	**371.75**	1.39	1.33	1.30	1.18	1.17	1.11
Quadruped	**959.75**	**958.44**	**829.41**	206.24	178.01	135.29	214.13	208.47	202.72
Trading	**37.72**	**33.09**	**31.98**	9.17	-3.36	-4.77	12.76	11.94	4.19

(c) CT-SAC versus continuous-time algorithms CPPO, q-Learning, and CT-TD3

	CT-SAC			CPPO			q-Learning			CT-TD3		
Task	**top**	**second**	**third**	**top**	**second**	**third**	**top**	**second**	**third**	**top**	**second**	**third**
Cheetah	**934.76**	**863.45**	**807.19**	174.50	168.64	145.92	348.61	328.69	272.00	705.86	457.24	268.61
Walker	**1035.52**	**928.26**	**817.28**	51.77	44.24	42.04	119.78	94.79	74.67	596.50	578.19	567.22
Humanoid	**386.75**	**379.75**	**371.75**	1.16	1.08	1.04	1.81	1.59	1.28	337.23	326.61	295.94
Quadruped	**959.75**	**958.44**	**829.41**	160.19	159.80	159.05	339.49	299.32	216.92	484.25	455.86	345.87
Trading	**37.72**	**33.09**	**31.98**	23.46	23.37	22.32	12.22	2.37	-39.94	6.53	-0.11	-6.22

B.3 Duration-aware adaptations of SAC

We further test whether standard SAC can account for irregular decision times through adaptations that **incorporate elapsed time into reward scaling and discounting**. We consider two variants. *SAC-increment modeling* rescales rewards according to:

$$r_t^{\text{new}} = \Delta_t r_t, \tag{31}$$

while retaining the standard fixed discount factor. *SAC-time-aware* additionally makes the continuation discount depend on elapsed time,

$$r_t^{\mathrm{TA}} = \Delta_t r_t, \qquad \gamma_t = e^{-\beta \Delta_t}. \tag{32}$$

Thus, both baselines explicitly incorporate the observed transition duration while retaining the standard SAC critic structure, with SAC-time-aware accounting for elapsed time in both reward scaling and discounting.

As shown in Figures 4 and 5, neither adaptation consistently improves over standard SAC, let alone closes the gap to CT-SAC. SAC-time-aware only approaches the final SAC performance on Cheetah and performs substantially worse on Walker, Humanoid, Quadruped, and Trading, while increment modeling likewise provides no consistent benefit. Although standard SAC eventually reaches a final return comparable to CT-SAC on Walker, CT-SAC learns substantially faster. Overall, these results show that adapting discrete-time SAC to variable time steps through reward scaling and duration-aware discounting is not sufficient. Hence, robust learning under irregular decision times benefits from a formulation designed directly for continuous time, as in CT-SAC.

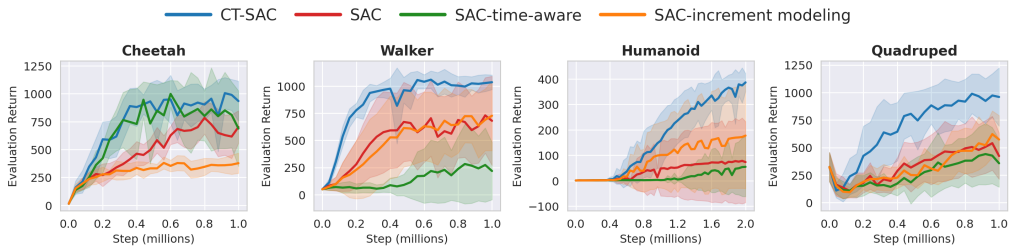


Figure 4: Evaluation returns for CT-SAC, SAC, SAC-time-aware, and SAC-increment modeling on the irregular-time control tasks over 12 seeds.

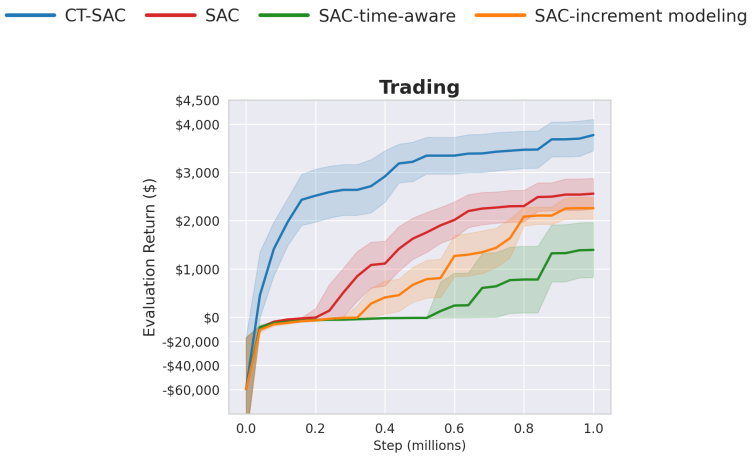


Figure 5: Evaluation returns for CT-SAC, SAC, SAC-time-aware, and SAC-increment modeling on the trading task over 12 seeds.

B.4 Performance of all algorithms

To facilitate comparison across all eight algorithms, Figure 6 reports the **mean evaluation return only** (without variance bands). This view highlights the overall ranking and learning dynamics without clutter; variance plots and full per-seed statistics are already provided in the main paper and in the appendix.

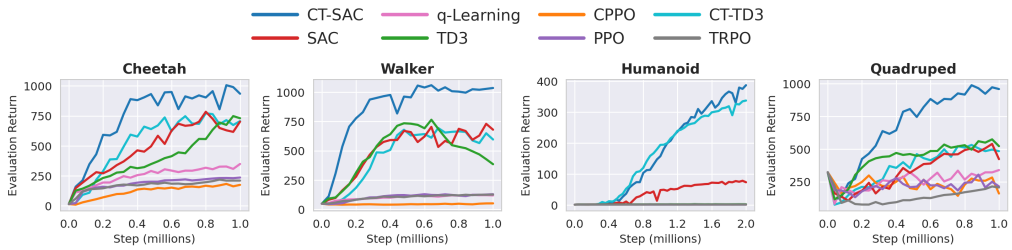


Figure 6: Evaluation returns for all algorithms on the control task over 12 seeds.

B.5 Runtime and time complexity

Runtime. We summarize the runtime in Table 3, reported as mean $\pm$ std over 12 seeds for the **top** configuration of each algorithm. Overall, **CT-SAC has runtime comparable to SAC**, and **CT-TD3 is comparable to TD3**, indicating that our continuous-time modifications do not materially increase training cost relative to their discrete-time counterparts.

Time complexity. CT-SAC follows the same training structure as SAC: sampling rollouts and performing batched critic/actor updates at a fixed update schedule. Let N denote the number of environment interaction steps (or sampled decision times), f the update frequency (updates every f steps), and d the number of gradient updates per update event. Then the total number of update steps scales as $\mathcal{O}(Nd/f)$. Per update, CT-SAC and SAC share the same dominant costs (forward/backward passes through the actor/critic networks). CT-SAC additionally includes time operations (e.g., dividing by Δ_t or using Δ_t in targets), which may increase constants slightly but does not change asymptotic complexity.

Table 3: Training runtime (hours) for each algorithm-task pair, reported as mean$\pm$std over 12 seeds.

Task	CT-SAC	SAC	TD3	TRPO	PPO	CPPO	q-Learning	CT-TD3
Cheetah	10.62 ± 0.53	6.29 ± 0.04	3.98 ± 0.05	0.44 ± 0.07	$\mathbf{0.29 \pm 0.07}$	0.57 ± 0.06	9.12 ± 0.17	4.60 ± 0.04
Walker	9.69 ± 0.83	8.90 ± 0.54	4.57 ± 0.24	$\mathbf{0.40 \pm 0.00}$	0.41 ± 0.07	0.61 ± 0.04	5.59 ± 0.08	4.68 ± 0.09
Humanoid	8.30 ± 0.50	6.02 ± 0.09	3.87 ± 0.06	0.82 ± 0.01	$\mathbf{0.48 \pm 0.02}$	1.01 ± 0.02	5.07 ± 0.14	5.32 ± 0.16
Quadruped	8.21 ± 0.41	6.49 ± 0.04	4.66 ± 0.04	0.40 ± 0.02	0.41 ± 0.01	$\mathbf{0.36 \pm 0.01}$	6.87 ± 0.32	2.65 ± 1.91
Trading	8.42 ± 0.11	7.42 ± 0.07	4.90 ± 0.08	0.81 ± 0.00	$\mathbf{0.52 \pm 0.01}$	0.89 ± 0.17	6.38 ± 0.10	5.44 ± 0.08

B.6 Hyperparameters tuning

We perform hyperparameter search to select the best configuration for each algorithm–task pair. Because the full sweep spans 8 algorithms $\times$ 5 tasks with distinct search spaces (in total roughly 40 algorithm–task search families), we place the complete search ranges and selected **top/second/third** settings in the codebase under the `out/final_reports/hyperparam_spaces` folder.

B.7 Statistical testing

We report statistical comparisons across 12 independent seeds. We use the same seeds across all algorithms and tasks. We run two complementary tests using `scipy.stats`: Welch's t-test and a paired test. Across tasks, these tests support that **CT-SAC outperforms the baselines** with statistically significant differences in the majority of settings. The full set of p-values and other details are included in `out/final_reports/significant_testing.csv` (available in the codebase repository).

B.8 Codebase and reproducibility

Our codebase is available at the link: `https://anonymous.4open.science/r/ct-rl`. Additional plots/media/tables are under `out/final_reports` folder.

Because training logs and saved checkpoints are large (∼6 GB), we provide them via the SwissTransfer links, given in the codebase: `https://anonymous.4open.science/r/ct-rl`. A step-by-step reproduction guide (exact commands, dependencies, seeds, and evaluation protocol) is documented in `README.md`, including how to regenerate the tables and figures from raw logs.

B.9 Generalization from irregular-time training to regular-time evaluation

We further evaluate whether policies trained under *irregular* decision times transfer to the standard *regular*-time evaluation setting. This test checks whether the learned policies capture the underlying control structure rather than overfitting to a particular irregular sampling pattern. Table 4 reports mean evaluation return over 12 seeds on the control tasks. CT-SAC achieve the best performance on all tasks, followed closely by CT-TD3. This indicates strong generalization from complex, noisy, intertwined irregular-time trajectories to the regular-time setting.

Table 4: Regular-time evaluation of policies trained under irregular time steps. We report mean return over 12 seeds on control tasks.

Task	CT-SAC	SAC	TD3	PPO	TRPO	q-Learning	CPPO	CT-TD3
Cheetah	**605.05**	**455.74**	404.18	124.45	104.46	208.04	102.03	**513.39**
Walker	**610.03**	403.44	**509.23**	73.06	64.55	70.94	25.04	**512.30**
Humanoid	**404.64**	**77.14**	2.27	1.34	1.35	1.90	1.42	**356.90**
Quadruped	**530.13**	**328.30**	**325.87**	159.52	121.84	192.86	125.30	284.23

Finally, we provide additional visualization for the trading task (Figure 7) and for control tasks under both the irregular setting (Figures 8 and 9) and the regular setting (Figures 10 and 11). Further implementation and evaluation details are provided in Section J.

C Preliminaries for theoretical analysis

State, action, and function space. Let $\mathcal{S} \subseteq \mathbb{R}^d$ be the state space and let $\mathcal{A}$ be a compact metric action space. For value functions, we work on the value function space $\mathcal{D} \subset C_b(\mathbb{R}^d)$ of bounded continuous functions equipped with the sup-norm $\|V\|_\infty := \sup_{x\in\mathbb{R}^d} |V(x)|$.

For convenience, we recall all notations and definitions introduced in Section 2, Section 3, and Section 4 in the main paper.

C.1 Notation

Table 5: Notations used throughout the paper.

Symbol	Meaning
$V^{(\alpha)}(x)$	Optimal value function with temperature α.
$V(x)$	Generic value function (not necessarily optimal).
$q_V(x,a)$	Advantage-rate q-function associated with V at state–action (x,a).
$q_V^u(x,a)$	Finite-time approximation of $q_V(x,a)$ using time increment u.
$\tilde{q}_V^u(x,a)$	Richardson-interpolated version of $q_V^u(x,a)$.
L^a	Infinitesimal generator under fixed action a.
$H_a(V)$	Action-wise Hamiltonian at a, defined via L^aV.
$H^{(0)}(V)$	Hard Hamiltonian (standard max aggregation over actions).
$H^{(\alpha)}(V)$	Soft Hamiltonian with temperature α.
$H_u^{(\alpha)}(V)$	Soft Hamiltonian using the finite-time approximation q_V^u.
$\tilde{H}_u^{(\alpha)}(V)$	Soft Hamiltonian using the Richardson approximation $\tilde{q}_V^u$.
$\mathcal{Q}^{(\alpha)}$	Soft aggregation operator acting on q-functions (temperature α).
$Q(x,a)$	Numerical Q-function defined by $Q := V + q_V$.

Symbol	Meaning
$T_\tau^{(\alpha)}(V)$	Picard–Hamiltonian operator: $T_\tau^{(\alpha)}(V) := V + \tau H^{(\alpha)}(V)$.
$\Phi_\tau^{(\alpha)}$	Probabilistic dynamic-programming semigroup (step size τ, temperature α).
X_t	State process at time t.
a_t	Action at time t.
W_t	Brownian motion at time t.
b, σ	Drift and diffusion coefficients of the controlled SDE.
$\tilde{b}, \tilde{\sigma}, \tilde{X}_t$	Policy-averaged drift/diffusion and the induced averaged state process.
L^π	Policy-averaged generator: $L^\pi = \mathbb{E}_{a\sim\pi(\cdot\|x)}[L^a]$.
P_t^a	Markov semigroup of the diffusion under fixed action a.
$\mathcal{F}_{\tau,u}$	Single-critic update operator for the Q-function (step size τ, increment u).
$\mathcal{F}_{\tau,u}^{\mathrm{Rich}}$	Richardson version of the single-critic update operator $\mathcal{F}_{\tau,u}$.
$\bar{H}_U^{(\alpha)}(V)$	Random-increment Hamiltonian: $\bar{H}_U^{(\alpha)}(V) = \mathbb{E}[H_U^{(\alpha)}(V)]$ for random U.
$\tilde{T}_\tau^{(\alpha)}(V)$	Random-time Picard–Hamiltonian operator corresponding to $\bar{H}_U^{(\alpha)}(V)$.

C.2 Controlled diffusion dynamics

Given a progressively measurable control process $(a_t)_{t\geq 0}$ taking values in $\mathcal{A}$, we consider the controlled stochastic differential equation (SDE)

$$dX_t = b(X_t, a_t)\,dt + \sigma(X_t, a_t)\,dW_t, \qquad X_0 = x \in \mathbb{R}^d, \tag{33}$$

where W_t is a d-dimensional Brownian motion and $b : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^{d\times d}$ are measurable.

Assumption C.1 (Dynamics regularity)**.** The drift b and diffusion σ satisfy:

1. **Global Lipschitz and linear growth.** There exists $C > 0$ such that for all $x, y \in \mathbb{R}^d$ and $a \in \mathcal{A}$ we have:

$$\|b(x,a)-b(y,a)\|+\|\sigma(x,a)-\sigma(y,a)\| \leq C\|x-y\|, \quad \|b(x,a)\|+\|\sigma(x,a)\| \leq C\,(1+\|x\|).$$

2. **Uniform nondegeneracy and boundedness.** There exist $0 < \underline{\sigma} \leq \overline{\sigma} < \infty$ such that

$$\underline{\sigma}^2 I \preceq \sigma(x,a)\sigma(x,a)^\top \preceq \overline{\sigma}^2 I \qquad \forall x \in \mathbb{R}^d,\ \forall a \in \mathcal{A}.$$

Assumption C.2 (Reward and policy regularity)**.** The running reward $r : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ satisfies:

1. $r(\cdot, a)$ is bounded and is globally Lipschitz uniformly in a, i.e., there exist $R_{\max} < \infty$ and $L_r < \infty$ such that for all x, y and all a,

$$|r(x,a)| \leq R_{\max}, \qquad |r(x,a) - r(y,a)| \leq L_r\|x-y\|$$

2. For the entropy term, admissible Markov policies $\pi(\cdot \mid x)$ satisfy that $\log \pi(a \mid x)$ is well-defined and uniformly integrable under the induced state-action distribution.

Assumption C.3 (Regularity of value functions)**.** Suppose that the function class $\mathcal{D} \subset C_b^2(\mathbb{R}^d)$ of value functions satisfies the following Lipschitz conditions:

1. For each $V \in \mathcal{D}$ and $a \in \mathcal{A}$, the function L^aV is globally Lipschitz in x, uniformly in a, i.e., there exists the global constant $C_\mathcal{D}$ depending only on the value function class $\mathcal{D}$ such that L^aV is Lipschitz with the Lipschitz coefficient $C_\mathcal{D}$ for each $a \in \mathcal{A}$.

2. Additionally, for $\alpha > 0$, we assume that policies π defined by $x \mapsto \log \pi(a \mid x)$ is also globally Lipschitz uniformly in a with Lipschitz constant $C'_\mathcal{D}$.

Here the function class $\mathcal{D}$ refers to the domain of all value functions V considered throughout the theoretical analysis. Note that in the implementation, $\mathcal{D}$ is the value function class consisting of neural networks function with bounded weights/biases and smooth activation. As a result, this function class $\mathcal{D}$ naturally satisfies assumption C.3.

C.3 Policies and value functions

Fix a discount $\beta > 0$ and an entropy coefficient $\alpha \geq 0$. A Markov policy π maps each state x to a probability distribution $\pi(\cdot \mid x)$ on $\mathcal{A}$, and we write $\mathbb{E}_x^\pi[\cdot]$ for expectation under (33) when $a_t \sim \pi(\cdot \mid X_t)$. Recall that the entropy-regularized objective of policy π is:

$$V^{(\alpha)}(x;\pi) := \mathbb{E}_x^\pi\left[\int_0^\infty e^{-\beta t}\Big(r(X_t,a_t) - \alpha \log \pi(a_t \mid X_t)\Big)\, dt\right], \tag{34}$$

and the optimal value function is $V^{(\alpha)}(x) := \sup_\pi V^{(\alpha)}(x;\pi)$. The case $\alpha = 0$ recovers the standard (non-regularized) value function. Additionally, for notational convenience, if π and x is clear from the contex, we simply use $\mathbb{E}[.]$.

C.4 Infinitesimal generator and q-functions

Recall the following notations from Section 2 and Section 3. For a fixed action $a \in \mathcal{A}$, the controlled infinitesimal generator L^a acting on $\varphi \in C_b^2(\mathbb{R}^d)$ is:

$$(L^a\varphi)(x) := b(x,a)\cdot\nabla\varphi(x) + \frac{1}{2}\operatorname{Tr}\Big(\sigma(x,a)\sigma(x,a)^\top \nabla^2\varphi(x)\Big). \tag{35}$$

Given $V \in \mathcal{D}$, the q-function has the form:

$$q_V(x,a) := r(x,a) + (L^aV)(x) - \beta V(x). \tag{36}$$

For a short horizon $u > 0$, the finite time estimation for q under constant action a is:

$$q_V^u(x,a) := \frac{e^{-\beta u}\,\mathbb{E}_x^a[V(X_u)] - V(x)}{u} + r(x,a), \tag{37}$$

where $\mathbb{E}_x^a[\cdot]$ denotes expectation for the SDE (33) with constant control $a_t \equiv a$.

C.5 Dynkin's formula and small-time moments

Lemma C.4 (Dynkin's formula for controlled diffusions)**.** *Let $(X_t)_{t\geq 0}$ solve* (33) *under a progressively measurable control $(a_t)_{t\geq 0}$ and let τ be a bounded stopping time. Then for any $f \in C_b^2(\mathbb{R}^d)$,*

$$\mathbb{E}\big[f(X_\tau)\big] = f(x) + \mathbb{E}\left[\int_0^\tau (L^{a_t}f)(X_t)\, dt\right], \tag{38}$$

where $(L^{a_t}f)(X_t)$ is understood pointwise using (35)*. Additionally, under the diffusion without control: $dX_t = b(X_t)\,dt + \sigma(X_t)\,dW_t$, Equation* (38) *also holds with the (uncontrolled) infinitesimal generator $L\varphi(x) := b(x)\cdot\nabla\varphi(x) + \frac{1}{2}\operatorname{Tr}\Big(\sigma(x)\sigma(x)^\top\nabla^2\varphi(x)\Big)$. This can easily be seen by treating drift $b(x)$ as $b(x,a_0)$, and diffusion $\sigma(x)$ as $\sigma(x,a_0)$ for a fixed constant action a_0.*

Proof. This is a direct consequence of Itô's formula applied to $f(X_t)$ and taking expectations [5]. □

We now recall a standard result in stochastic processes.

Lemma C.5 (Small-time moment bounds)**.** *Under Assumption C.1, for X_t in Equation* (33)*, there exists a constant $C > 0$ such that for all $t \in (0,1]$ and for any admissible control process $(a_s)_{0\leq s\leq t}$,*

$$\mathbb{E}_x\big[\|X_t - x\|\big] \leq C\sqrt{t}, \tag{39}$$

and moreover

$$\mathbb{E}_x\left[\sup_{0\leq s\leq t}\|X_s - x\|^2\right] \leq Ct, \qquad \mathbb{E}_x\left[\sup_{0\leq s\leq t}\|X_s - x\|\right] \leq C\sqrt{t}. \tag{40}$$

The constant C depends only on the constants in Assumption C.1 and d, and is independent of the choice of control.

Proof sketch. We have $X_t - x = \int_0^t b(X_s, a_t)\, du + \int_0^s \sigma(X_t, a_t)\, dW_u$. Because of the linear growth, we get $\|b(x,a)\|^2 \le K(1+\|x\|^2)$ and $\|\sigma(x,a)\|_F^2 \le K(1+\|x\|^2)$ for some constant K. From here, we bound the second moment through Itô isometry and Cauchy-Schwarz inequalities:

$$\begin{aligned}\mathbb{E}\|X_t - x\|^2 &\le 2\mathbb{E}\Big\| \int_0^t b(X_s, a_s)\, ds\Big\|^2 + 2\mathbb{E}\Big\| \int_0^t \sigma(X_s, a_s)\, dW_s\Big\|^2 \\ &\le 2t \int_0^t \mathbb{E}\|b(X_s, a_s)\|^2 ds + 2\int_0^t \mathbb{E}\|\sigma(X_s, a_s)\|_F^2 ds \\ &\le 2t \int_0^t K(1+\|X_s\|^2) ds + 2\int_0^t K(1+\|X_s\|^2) ds \le C_1 t,\end{aligned}$$

as we can bound $\|X_s\|^2$ easily through an argument using Gronwall's lemma. Hence, by Cauchy-Schwarz again, we obtain the desired bound. For the supremum, note that with $A_s := \int_0^s b(X_u, a_u)\, du$ and $M_s := \int_0^s \sigma(X_u, a_u)\, dW_u$, we have

$$\sup_{0\le s\le t} \|X_s - x\|^2 \le 2\sup_{s\le t}\|A_s\|^2 + 2\sup_{s\le t}\|M_s\|^2,$$

We can then apply the Burkholder–Davis–Gundy inequality to the martingale part M_t and use similar estimates to obtain desired bound. □

C.6 Stochastic control under relaxed (policy-averaged) dynamics

When the action is sampled from a Markov policy $\pi(\cdot \mid X_t)$, it is convenient to use an equivalent diffusion with drift and covariance averaged under π. For a fixed Markov policy π, define:

$$\tilde{b}(x;\pi) := \int_{\mathcal{A}} b(x,a)\, \pi(da \mid x), \tag{41}$$

Here $\pi(da|x)$ is the measured induced by the policy $\pi(.|x)$ on $\mathcal{A}$. We will use this notation rather than $\int_{a\in\pi(.|x)} b(x,a)da$ from now on, as it is more aligned with the measure-theoretic nature of the theoretical details. We also define $\tilde{\sigma}(x;\pi)$ via the covariance identity

$$\tilde{\sigma}(x;\pi)\tilde{\sigma}(x;\pi)^\top := \int_{\mathcal{A}} \sigma(x,a)\sigma(x,a)^\top\, \pi(da \mid x). \tag{42}$$

Under Assumption C.1 and the mild Lipschitz regularity of π in Assumption C.3, there exists a filtered probability space supporting a Brownian motion $\tilde{W}$ and a process $(\tilde{X}_t)_{t\ge 0}$ solving

$$d\tilde{X}_t = \tilde{b}(\tilde{X}_t;\pi)\, dt + \tilde{\sigma}(\tilde{X}_t;\pi)\, d\tilde{W}_t, \qquad \tilde{X}_0 = x, \tag{43}$$

For each $t \ge 0$, [11; 28] state that the law of $\tilde{X}_t$ coincides with the law of X_t under the original controlled system (33) for $a_t \sim \pi(.|X_t)$. Note that, for the theoretical analysis, we work with the policy-averaged version in (43). The original controlled formulation is retained only to make the connection with discrete-time MDPs explicit. Equivalently, one may start directly from (43), which can be viewed as the continuous-time limit of the randomized controlled dynamics in (33), with actions sampled from $\pi(\cdot \mid X_t)$ and held piecewise constant between decision times.

For $\varphi \in C_b^2(\mathbb{R}^d)$, we also define relaxed infinitesimal generator with two equivalent representations:

$$(L^\pi\varphi)(x) := \int_{\mathcal{A}} (L^a\varphi)(x)\, \pi(da \mid x) = \tilde{b}(x;\pi)\cdot\nabla\varphi(x) + \frac{1}{2}\operatorname{Tr}\Big(\tilde{\sigma}(x;\pi)\tilde{\sigma}(x;\pi)^\top \nabla^2\varphi(x)\Big). \tag{44}$$

Finally, we may write the policy value (34) equivalently as

$$V^{(\alpha)}(x;\pi) = \mathbb{E}\left[\int_0^\infty e^{-\beta t}\, \tilde{r}^{(\alpha)}(\tilde{X}_t;\pi)\, dt \,\middle|\, \tilde{X}_0 = x\right], \tag{45}$$

where the relaxed running reward is

$$\tilde{r}^{(\alpha)}(x;\pi) := \int_{\mathcal{A}} \Big(r(x,a) - \alpha \log \pi(a \mid x)\Big)\, \pi(da \mid x). \tag{46}$$

Note that the drift function $\tilde{b}$ defined by averaging over policy randomness is similar to the result obtained by [15] that we mentioned in Section 2.1. Here we also have the diffusion (Brownian motion) part. See [25; 23; 11; 28] for more details on the policy-averaged SDE.

Lastly, we would like to state an useful lemma that will be used throughout Appendices D, E, and F.

Lemma C.6 (Entropy-regularized expectation via a KL shift)**.** *Fix $\alpha > 0$ and let $q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ be any measurable function. For each $x \in \mathcal{X}$ recall the functional $\mathcal{Q}$ on q (see Section 3.2) that:*

$$\mathcal{Q}^{(\alpha)}(q)(x) = \alpha \log \left(\int_{\mathcal{A}} \exp\Big(q(x,a)/\alpha\Big)\, da \right) = \alpha \log Z_q(x),$$

with partition function Z_q, and the Boltzmann distribution π_q:

$$\pi_q(a \mid x) := \frac{\exp(q(x,a)/\alpha)}{Z_q(x)}, \quad Z_q(x) := \int_{\mathcal{A}} \exp\Big(q(x,a)/\alpha\Big)\, da.$$

Then for any policy $\pi(\cdot \mid x)$,

$$\alpha \log Z_q(x) - \alpha\, \mathrm{KL}\big(\pi(\cdot \mid x) \,\|\, \pi_q(\cdot \mid x)\big) = \mathbb{E}_{a\sim\pi(\cdot|x)}\big[q(x,a) - \alpha \log \pi(a \mid x)\big]. \tag{47}$$

In particular,

$$\mathcal{Q}^{(\alpha)}(q)(x) = \alpha \log Z_q(x) = \sup_{\pi(\cdot|x)} \mathbb{E}_{a\sim\pi(\cdot|x)}\big[q(x,a) - \alpha \log \pi(a \mid x)\big], \tag{48}$$

and the supremum is achieved uniquely by $\pi = \pi_q$.

Proof. By definition,

$$\mathrm{KL}(\pi \| \pi_q) = \int_{\mathcal{A}} \pi(a \mid x) \log \frac{\pi(a \mid x)}{\pi_q(a \mid x)}\, da = \int_{\mathcal{A}} \pi(a \mid x) \log \pi(a \mid x)\, da - \int_{\mathcal{A}} \pi(a \mid x) \log \pi_q(a \mid x)\, da.$$

Since $\log \pi_q(a \mid x) = \frac{1}{\alpha} q(x,a) - \log Z_q(x)$, we obtain

$$\mathrm{KL}(\pi \| \pi_q) = \int_{\mathcal{A}} \pi(a \mid x) \log \pi(a \mid x)\, da - \frac{1}{\alpha} \mathbb{E}_\pi[q(x,a)] + \log Z_q(x).$$

Rearranging terms gives us the equation (47). Finally, (48) follows because $\mathrm{KL}(\pi \| \pi_q) \geq 0$ with equality iff $\pi = \pi_q$. □

D Proof of value function convergence (Theorem 4.1)

This appendix proves the convergence result for the **ideal** Picard–Hamiltonian value iteration $V_{k+1} = T_\tau^{(\alpha)}(V_k)$ in Theorem 4.1. The key idea is to compare the Picard update to the short-horizon dynamic-programming operator $\Phi_\tau^{(\alpha)}$, which is a strict contraction in the sup-norm.

D.1 Dynamic-programming operator and contraction

For $\tau > 0$ and $\alpha \geq 0$, recall the entropy-regularized dynamic-programming operator $\Phi_\tau^{(\alpha)} : \mathcal{D} \to \mathcal{D}$ defined earlier:

$$(\Phi_\tau^{(\alpha)} V)(x) := \sup_\pi \mathbb{E}_x^\pi \left[\int_0^\tau e^{-\beta t} \Big(r(X_t, a_t) - \alpha \log \pi(a_t \mid X_t) \Big)\, dt + e^{-\beta\tau} V(X_\tau) \right], \tag{49}$$

where X follows the controlled SDE (33) and $a_t \sim \pi(\cdot \mid X_t)$.

Lemma D.1 (Contraction of $\Phi_\tau^{(\alpha)}$)**.** *Assume Assumption C.1. Then for all $\tau > 0$ and $\alpha \geq 0$,*

$$\|\Phi_\tau^{(\alpha)}(V) - \Phi_\tau^{(\alpha)}(W)\|_\infty \leq e^{-\beta\tau}\, \|V - W\|_\infty, \qquad \forall V, W \in X. \tag{50}$$

Consequently, $\Phi_\tau^{(\alpha)}$ admits a unique fixed point $V^{(\alpha)} \in X$ and

$$\|(\Phi_\tau^{(\alpha)})^{(k)}(V_0) - V^{(\alpha)}\|_\infty \leq e^{-\beta\tau k}\, \|V_0 - V^{(\alpha)}\|_\infty, \qquad \forall k \geq 0. \tag{51}$$

Proof. Fix $x \in \mathbb{R}^d$ and any policy π. Since the running reward terms cancel,

$$\left|\mathbb{E}_x^\pi\left[\int_0^\tau e^{-\beta t}(r-\alpha\log\pi)\,dt + e^{-\beta\tau}V(X_\tau)\right] - \mathbb{E}_x^\pi\left[\int_0^\tau e^{-\beta t}(r-\alpha\log\pi)\,dt + e^{-\beta\tau}W(X_\tau)\right]\right|$$
$$= e^{-\beta\tau}\left|\mathbb{E}_x^\pi\big[V(X_\tau) - W(X_\tau)\big]\right| \le e^{-\beta\tau}\,\|V-W\|_\infty.$$

so $\Phi_\tau^{(\alpha)}$ is a contraction on bounded functions with modulus $e^{-\beta\tau} < 1$. By the Banach fixed-point theorem, it admits a unique fixed point $V^\star$, and $(\Phi_\tau^{(\alpha)})^{(k)}(V_0)$ converge to $V^\star$ at a geometric rate. Finally, the entropy-regularized value function $V^{(\alpha)}$ satisfies the dynamic programming principle, hence $V^{(\alpha)}$ is a fixed point of $\Phi_\tau^{(\alpha)}$. By uniqueness of the fixed point, $V^\star = V^{(\alpha)}$, and Equation (51) follows trivially. □

D.2 Picard–Hamiltonian operator

Recall the infinitesimal rate q_V from (36):

$$q_V(x,a) = r(x,a) + (L^aV)(x) - \beta V(x),$$

and the (soft/hard) Hamiltonian $H^{(\alpha)}(V) : \mathbb{R}^d \to \mathbb{R}$:

$$H^{(\alpha)}(V)(x) := \begin{cases} \alpha\log\displaystyle\int_{\mathcal{A}} \exp\Big(\frac{q_V(x,a)}{\alpha}\Big)\,da, & \alpha > 0, \\ \sup\limits_{a\in\mathcal{A}} q_V(x,a), & \alpha = 0, \end{cases} \tag{52}$$

The **ideal** Picard–Hamiltonian operator has the form:

$$(T_\tau^{(\alpha)}V)(x) := V(x) + \tau\, H^{(\alpha)}(V)(x). \tag{53}$$

D.3 Small-time expansions

We first give a small-time expansion for a *fixed action* a and then for a *fixed policy* π. The policy expansion is the missing ingredient needed to justify the per-step mismatch bound for $\alpha > 0$.

Lemma D.2 (Small-time expansion, fixed action)**.** *Assume Assumptions C.1, C.2, and C.3. Fix $a \in \mathcal{A}$ and $V \in \mathcal{D}$. Define*

$$(\Phi_\tau^a V)(x) := \mathbb{E}_x^a\left[\int_0^\tau e^{-\beta t} r(X_t,a)\,dt + e^{-\beta\tau}V(X_\tau)\right], \tag{54}$$

where X_t follows (33) *with constant control $a_t \equiv a$. Then for all sufficiently small $\tau \in (0,1]$,*

$$(\Phi_\tau^a V)(x) = V(x) + \tau\, q_V(x,a) + R_\tau^a(V)(x), \tag{55}$$

and the remainder satisfies

$$\sup_{x\in\mathbb{R}^d,\, a\in\mathcal{A}} |R_\tau^a(V)(x)| \le C\,\tau^{3/2}, \tag{56}$$

where C depends on bounds/Lipschitz constants of b, σ, r and on $C_{\mathcal{D}}$ only but is independent of τ.

Proof. We treat the terminal and running reward parts separately.

Terminal term. By Dynkin's formula (Lemma C.4) under constant action a,

$$\mathbb{E}_x^a[V(X_\tau)] - V(x) = \int_0^\tau \mathbb{E}_x^a\big[(L^aV)(X_t)\big]\,dt.$$

Add and then subtract $(L^aV)(x)$ inside the integral gives:

$$\mathbb{E}_x^a[V(X_\tau)] = V(x) + \tau(L^aV)(x) + \int_0^\tau \mathbb{E}_x^a\Big[(L^aV)(X_t) - (L^aV)(x)\Big]\,dt.$$

Since L^aV is globally Lipschitz in x with constant $C_{\mathcal{D}}$ (Assumption C.3), Lemma C.5 implies that:

$$\Big|\int_0^\tau \mathbb{E}_x^a\big[(L^aV)(X_t) - (L^aV)(x)\big]\,dt\Big| \le C\int_0^\tau \sqrt{t}\,dt = \mathcal{O}(\tau^{3/2}).$$

Therefore,

$$\mathbb{E}_x^a[e^{-\beta\tau}V(X_\tau)] = V(x) + \tau\big((L^aV)(x) - \beta V(x)\big) + \mathcal{O}(\tau^{3/2}),$$

using $e^{-\beta\tau} = 1 - \beta\tau + \mathcal{O}(\tau^2)$ and absorbing $\mathcal{O}(\tau^2)$ into $\mathcal{O}(\tau^{3/2})$.

Running reward term. Using Lipschitz continuity of $r(\cdot, a)$ and Lemma C.5,

$$\Big|\mathbb{E}_x^a\Big[\int_0^\tau e^{-\beta t} r(X_t, a)\, dt\Big] - r(x,a)\int_0^\tau e^{-\beta t}\, dt\Big| \le \int_0^\tau e^{-\beta t}\, L_r\, \mathbb{E}_x^a\|X_t - x\|\, dt \le C\int_0^\tau \sqrt{t}\, dt = \mathcal{O}(\tau^{3/2}).$$

Since $\int_0^\tau e^{-\beta t}dt = \tau + \mathcal{O}(\tau^2)$, we conclude

$$\mathbb{E}_x^a\Big[\int_0^\tau e^{-\beta t} r(X_t, a)\, dt\Big] = \tau r(x,a) + \mathcal{O}(\tau^{3/2}).$$

Combining the two expansions and the definition of q-function ($q_V(x,a) = r(x,a) + (L^aV)(x) - \beta V(x)$) yields (55) and the remainder bound (56). □

Lemma D.3 (Small-time expansion, fixed policy via relaxed dynamics)**.** *Assume Assumptions C.1, C.2, and C.3. Fix a Markov policy π and $V \in \mathcal{D}$. Let $(\tilde{X}_t)_{t\ge 0}$ follow the relaxed SDE* (43) *and define the relaxed reward $\tilde{r}^{(\alpha)}(\cdot;\pi)$ as in* (46)*. Define the fixed-policy short-horizon operator*

$$(\Phi_\tau^{\pi,(\alpha)}V)(x) := \mathbb{E}\Big[\int_0^\tau e^{-\beta t}\, \tilde{r}^{(\alpha)}(\tilde{X}_t;\pi)\, dt + e^{-\beta\tau}V(\tilde{X}_\tau)\,\Big|\,\tilde{X}_0 = x\Big]. \tag{57}$$

Then for all sufficiently small $\tau \in (0,1]$,

$$(\Phi_\tau^{\pi,(\alpha)}V)(x) = V(x) + \tau\Big(\tilde{r}^{(\alpha)}(x;\pi) + (L^\pi V)(x) - \beta V(x)\Big) + R_\tau^{\pi,(\alpha)}(V)(x), \tag{58}$$

where L^π is the relaxed generator in Equation (44)*, and*

$$\sup_{x\in\mathbb{R}^d} |R_\tau^{\pi,(\alpha)}(V)(x)| \le C_\alpha\, \tau^{3/2}, \tag{59}$$

with C_α independent of τ and uniform over π in the admissible class.

Proof. We again treat the terminal and running terms separately, similar to Lemma D.2.

Terminal term. Applying Dynkin's formula (Lemma C.4) to $V(\tilde{X}_t)$ under the relaxed generator L^π,

$$\mathbb{E}[V(\tilde{X}_\tau)] - V(x) = \int_0^\tau \mathbb{E}\big[(L^\pi V)(\tilde{X}_t)\big]\, dt.$$

Add and subtract $(L^\pi V)(x)$:

$$\mathbb{E}[V(\tilde{X}_\tau)] = V(x) + \tau(L^\pi V)(x) + \int_0^\tau \mathbb{E}\big[(L^\pi V)(\tilde{X}_t) - (L^\pi V)(x)\big]\, dt.$$

By Assumption C.3, $L^\pi V$ is globally Lipschitz in x with Lipschitz constant depending only on $C_{\mathcal{D}}$ and $C'_{\mathcal{D}}$ (using global Lipschitzness of L^aV and $\log\pi(\cdot\mid x)$). In addition, Lemma C.5 applies to $\tilde{X}$ as well, and hence the last integral is $\mathcal{O}(\tau^{3/2})$ uniformly in π. Multiplying by $e^{-\beta\tau} = 1-\beta\tau+\mathcal{O}(\tau^2)$ yields the estimate:

$$\mathbb{E}\big[e^{-\beta\tau}V(\tilde{X}_\tau)\big] = V(x) + \tau\big((L^\pi V)(x) - \beta V(x)\big) + \mathcal{O}(\tau^{3/2}).$$

Running term. Using Lipschitz continuity of $\tilde{r}^{(\alpha)}(\cdot;\pi)$ in x over the admissible class, together with Lemma C.5,

$$\Big|\mathbb{E}\Big[\int_0^\tau e^{-\beta t}\tilde{r}^{(\alpha)}(\tilde{X}_t;\pi)\, dt\Big] - \tilde{r}^{(\alpha)}(x;\pi)\int_0^\tau e^{-\beta t}\, dt\Big| \le \int_0^\tau e^{-\beta t}\, C\, \mathbb{E}\|\tilde{X}_t - x\|\, dt \le C\int_0^\tau \sqrt{t}\, dt = \mathcal{O}(\tau^{3/2}),$$

uniformly in π. Again, because $\int_0^\tau e^{-\beta t}dt = \tau + \mathcal{O}(\tau^2)$, we obtain:

$$\mathbb{E}\Big[\int_0^\tau e^{-\beta t}\tilde{r}^{(\alpha)}(\tilde{X}_t;\pi)\, dt\Big] = \tau\,\tilde{r}^{(\alpha)}(x;\pi) + \mathcal{O}(\tau^{3/2}).$$

Combining both parts gives (58) and (59). □

D.4 PER-STEP MISMATCH: $\Phi_\tau^{(\alpha)}$ VS. $T_\tau^{(\alpha)}$

Lemma D.4 (Per-step mismatch bound)**.** *Assume Assumptions C.1, C.2, and C.3. Let $V \in \mathcal{D}$ and $\alpha \geq 0$. Then for all sufficiently small $\tau \in (0, 1]$,*

$$\Phi_\tau^{(\alpha)}(V)(x) = V(x) + \tau\, H^{(\alpha)}(V)(x) + R_\tau^{(\alpha)}(V)(x), \tag{60}$$

with

$$\|R_\tau^{(\alpha)}(V)\|_\infty \leq C_\alpha\, \tau^{3/2}, \tag{61}$$

where C_α is independent of τ (and only depends on the universal constants of the assumptions). Consequently,

$$\|T_\tau^{(\alpha)}(V) - \Phi_\tau^{(\alpha)}(V)\|_\infty \leq C_\alpha\, \tau^{3/2}. \tag{62}$$

Proof. We treat $\alpha = 0$ and $\alpha > 0$ separately.

Case 1: $\alpha = 0$ (hard-max). For fixed a, Lemma D.2 yields

$$\Phi_\tau^a V(x) = V(x) + \tau q_V(x, a) + R_\tau^a(V)(x), \qquad \sup_{x,a} |R_\tau^a(V)(x)| \leq C\tau^{3/2}.$$

Since $\Phi_\tau^{(0)} V(x) = \sup_{a\in\mathcal{A}} \Phi_\tau^a V(x)$,

$$\begin{aligned}\Phi_\tau^{(0)} V(x) &= V(x) + \tau \sup_{a\in\mathcal{A}} q_V(x, a) + \sup_{a\in\mathcal{A}} R_\tau^a(V)(x) \\ &= V(x) + \tau H^{(0)}(V)(x) + R_\tau^{(0)}(V)(x),\end{aligned}$$

and $\|R_\tau^{(0)}(V)\|_\infty \leq C\tau^{3/2}$.

Case 2: $\alpha > 0$ (soft-max). For any fixed Markov policy π, Lemma D.3 gives

$$\Phi_\tau^{\pi,(\alpha)} V(x) = V(x) + \tau\Big(\tilde{r}^{(\alpha)}(x; \pi) + (L^\pi V)(x) - \beta V(x)\Big) + R_\tau^{\pi,(\alpha)}(V)(x),$$

with $\sup_x |R_\tau^{\pi,(\alpha)}(V)(x)| \leq C_\alpha \tau^{3/2}$ uniformly in π. Taking supremum over π and using $\Phi_\tau^{(\alpha)} V = \sup_\pi \Phi_\tau^{\pi,(\alpha)} V$ yields

$$\Phi_\tau^{(\alpha)} V(x) = V(x) + \tau \sup_\pi \Big(\tilde{r}^{(\alpha)}(x; \pi) + (L^\pi V)(x) - \beta V(x)\Big) + R_\tau^{(\alpha)}(V)(x),$$

where $\|R_\tau^{(\alpha)}(V)\|_\infty \leq C_\alpha \tau^{3/2}$.

It remains to identify the second supremum term as $H^{(\alpha)}(V)(x)$. Using the definitions of $\tilde{r}^{(\alpha)}$ and L^π,

$$\begin{aligned}\tilde{r}^{(\alpha)}(x; \pi) + (L^\pi V)(x) - \beta V(x) &= \int_{\mathcal{A}} \Big(r(x, a) + (L^a V)(x) - \beta V(x)\Big)\, \pi(da \mid x) \\ &\qquad - \alpha \int_{\mathcal{A}} \log \pi(a \mid x)\, \pi(da \mid x) \\ &= \int_{\mathcal{A}} q_V(x, a)\, \pi(da \mid x) - \alpha \int_{\mathcal{A}} \log \pi(a \mid x)\, \pi(da \mid x).\end{aligned}$$

Through Lemma C.6, for each fixed x,

$$\begin{aligned}&\sup_{\pi(\cdot|x)} \left\{ \int_{\mathcal{A}} q_V(x, a)\, \pi(da \mid x) - \alpha \int_{\mathcal{A}} \log \pi(a \mid x)\, \pi(da \mid x) \right\} \\ &= \sup_{\pi(\cdot|x)} \mathbb{E}_{a\sim\pi(\cdot|x)}\big[q_V(x, a) - \alpha \log \pi(a \mid x)\big] \\ &= \mathcal{Q}^{(\alpha)}(q_V)(x) = \alpha \log \int_{\mathcal{A}} \exp\Big(\frac{q_V(x, a)}{\alpha}\Big)\, da\end{aligned}$$

which is exactly $H^{(\alpha)}(V)(x)$ by (52). Substituting gives (60)–(61). Finally, (62) follows immediately from the definition $T_\tau^{(\alpha)}(V) = V + \tau H^{(\alpha)}(V)$. □

D.5 Proof of Theorem 4.1

Theorem 4.1 (Value Function Convergence). *Fix $\alpha \geq 0$ and $\tau > 0$. Under Assumptions C.1, C.2, and C.3, let V_k be the value function after k iterations $V_{k+1} := T_\tau^{(\alpha)}(V_k)$. Then, V_k converges to $V^{(\alpha)}$ as $\tau \to 0$, and for all $k \geq 0$,*

$$\|V_k - V^{(\alpha)}\|_\infty \leq C_0 e^{-\beta\tau k} + \frac{C_1 \tau^{3/2}}{1 - e^{-\beta\tau}}. \tag{30}$$

Here C_0 and C_1 are constants independent of k.

Proof. Let $W_{k+1} := \Phi_\tau^{(\alpha)}(W_k)$ be the exact dynamic-programming iterations initialized at the same $W_0 := V_0$. Define $\Delta_k := V_k - W_k$. Then

$$\begin{aligned}\Delta_{k+1} &= T_\tau^{(\alpha)}(V_k) - \Phi_\tau^{(\alpha)}(W_k) \\ &= \Big(T_\tau^{(\alpha)}(V_k) - \Phi_\tau^{(\alpha)}(V_k)\Big) + \Big(\Phi_\tau^{(\alpha)}(V_k) - \Phi_\tau^{(\alpha)}(W_k)\Big).\end{aligned}$$

Taking sup norms and using Lemma D.4 and Lemma D.1, we get:

$$\|\Delta_{k+1}\|_\infty \leq C_\alpha \tau^{3/2} + e^{-\beta\tau}\|\Delta_k\|_\infty.$$

Since $\Delta_0 = 0$, unrolling the recursion gives

$$\|\Delta_k\|_\infty \leq \frac{C_\alpha \tau^{3/2}}{1 - e^{-\beta\tau}} \qquad \forall k \geq 0.$$

Finally, by the triangle inequality and Lemma D.1,

$$\begin{aligned}\|V_k - V^{(\alpha)}\|_\infty &\leq \|V_k - W_k\|_\infty + \|W_k - V^{(\alpha)}\|_\infty \\ &\leq \frac{C_\alpha \tau^{3/2}}{1 - e^{-\beta\tau}} + e^{-\beta\tau k}\,\|W_0 - V^{(\alpha)}\|_\infty,\end{aligned}$$

which is the claimed bound. □

Note that, since $1 - e^{-\beta\tau} = \beta\tau + \mathcal{O}(\tau^2)$, the error estimate implies that $\left\|V_k - V^{(\alpha)}\right\| \leq C_0 e^{-\beta\tau k} + C_1 \tau^{1/2}$, and as $k \to \infty$, $\tau \to 0$, this error also goes to 0.

E Proof of q-convergence (Theorem 4.2)

This appendix proves the model-free convergence guarantee in Theorem 4.2. The key challenge is that the infinitesimal generator term $L^a V$ is unavailable in model-free learning, so we replace the infinitesimal advantage-rate q_V by a short-horizon estimator q_V^u computed from rollouts of length u (See Section 3.2). This introduces a *modelling bias* $q_V^u - q_V$ and an additional stability issue because q_V^u is $\mathcal{O}(1/u)$-Lipschitz in V. Richardson extrapolation removes the leading $\mathcal{O}(u)$ bias term and improves approximation accuracy of the learning signal while preserving the same stability structure.

E.1 Baseline modelling bias under standard assumptions

Before imposing higher-order smoothness (which yields $\mathcal{O}(u)$ and Richardson $\mathcal{O}(u^2)$), we record the baseline modelling-bias rate available under the standard diffusion assumptions. In this setting, one only obtains $\|q_V^u - q_V\|_\infty = \mathcal{O}(\sqrt{u})$.

Lemma E.1 (Baseline bias for q_V^u: $\|q_V^u - q_V\|_\infty = \mathcal{O}(\sqrt{u})$). *Assume Assumptions C.1 and C.2. Suppose $V \in C_b^2(\mathbb{R}^d)$ and $L^a V$ is globally Lipschitz in x uniformly over $a \in \mathcal{A}$. Then there exist $u_0 > 0$ and $C < \infty$ such that for all $u \in (0, u_0]$,*

$$\|q_V^u - q_V\|_\infty \leq C\sqrt{u}. \tag{63}$$

Consequently, for any $\alpha > 0$,

$$\|H_u^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \leq C\sqrt{u}. \tag{64}$$

Proof sketch. By Dynkin's formula,

$$\mathbb{E}_x^a[V(X_u^a)] - V(x) = \int_0^u \mathbb{E}_x^a[(L^a V)(X_t^a)]\, dt.$$

Subtract $u(L^aV)(x)$ and use the Lipschitz property of L^aV gives:

$$\left|\mathbb{E}_x^a[V(X_u^a)] - V(x) - u(L^aV)(x)\right| \le \int_0^u \mathbb{E}_x^a\big[\,|(L^aV)(X_t^a) - (L^aV)(x)|\,\big]dt \le C\int_0^u \mathbb{E}_x^a[\|X_t^a - x\|]dt.$$

Lemma C.5 give $\mathbb{E}_x^a[\|X_t^a - x\|] \le C\sqrt{t}$ uniformly in (x, a), and hence the right-hand side is $\mathcal{O}(u^{3/2})$. Through definitions of q_V^u and q_V, dividing by u and accounting for the discount factor $e^{-\beta u} = 1 - \beta u + \mathcal{O}(u^2)$ yields $q_V^u(x, a) = q_V(x, a) + \mathcal{O}(\sqrt{u})$ uniformly in (x, a), proving (63). The Hamiltonian bound (64) follows from a similar Lipschitz argument as in Lemma E.7. □

E.2 Smoothness assumptions for quantitative u-rates

Assumption E.2 (Smooth C^2 assumptions)**.** In addition to Assumption C.1, assume:

1. For each $a \in \mathcal{A}$, drift and diffusion maps $x \mapsto b(x, a)$ and $x \mapsto \sigma(x, a)$ are C^2 with bounded derivatives up to order 2, uniformly in a, and globally Lipschitz in x uniformly in a.
2. For each $a \in \mathcal{A}$, the reward map $x \mapsto r(x, a)$ is C^2 with bounded derivatives up to order 2, uniformly in a, and globally Lipschitz in x.

Assumption E.3 (Value function smoothness)**.** In addition to standard assumptions C.2 and C.3, assume that the value function domain $\mathcal{D} \subseteq C_b^4(\mathbb{R}^d)$ so that for value function V, and for each $a \in \mathcal{A}$, both L^aV and $(L^a)^2V$ are bounded and continuous.

Notation. Fix $a \in \mathcal{A}$. Let $(X_t^a)_{t\ge 0}$ solve the controlled SDE

$$dX_t^a = b(X_t^a, a)\, dt + \sigma(X_t^a, a)\, dW_t, \qquad X_0^a = x,$$

Recall the fixed action controlled generator L^a is:

$$(L^a f)(x) := b(x, a) \cdot \nabla f(x) + \frac{1}{2}\mathrm{Tr}\big(\sigma(x, a)\sigma(x, a)^\top \nabla^2 f(x)\big).$$

We additionally define the Markov semigroup

$$(P_t^a f)(x) := \mathbb{E}_x^a[f(X_t^a)]. \tag{65}$$

For $V \in C_b^2(\mathbb{R}^d)$, recall the q-function:

$$q_V(x, a) := r(x, a) + (L^aV)(x) - \beta V(x). \tag{66}$$

Together with its finite-step estimator q_V^u with time step $u > 0$:

$$q_V^u(x, a) := \frac{e^{-\beta u}\,\mathbb{E}_x^a\big[V(X_u^a)\big] - V(x)}{u} \;+\; r(x, a). \tag{67}$$

And the Richardson version $\tilde{q}_V^u$:

$$\tilde{q}_V^u(x, a) := 2\, q_V^{u/2}(x, a) - q_V^u(x, a). \tag{68}$$

We now start with a series of lemmas that serve as the main ingredients for the proof of Theorem 4.2.

E.3 Bias expansions for q_V^u and $\tilde{q}_V^u$

We begin with a quantitative small-time expansion of the semigroup P_t^a.

Lemma E.4 (Second-order semigroup expansion)**.** *Under assumptions E.2 and E.3, for $a \in \mathcal{A}$, $V \in C_b^4(\mathbb{R}^d)$, and for all $t \in [0,1]$, we have*

$$(P_t^a V)(x) = V(x) + t(L^a V)(x) + R_V(t;x,a), \tag{69}$$

where the remainder satisfies

$$\sup_{x\in\mathbb{R}^d,\, a\in\mathcal{A}} |R_V(t;x,a)| \le \frac{t^2}{2} \sup_{a\in\mathcal{A}} \|(L^a)^2 V\|_\infty \le C_V\, t^2, \tag{70}$$

for a constant $C_V < \infty$ depending only on V and the coefficients.

Proof. Fix a and write $P_t := P_t^a$ and $L := L^a$ for notation convenience. Dynkin's formula (Lemma C.4) gives

$$(P_t V)(x) = V(x) + \int_0^t (P_s LV)(x)\, ds.$$

Applying the same identity to LV one more time yields:

$$(P_s LV)(x) = (LV)(x) + \int_0^s (P_r L^2 V)(x)\, dr.$$

Substitute into the first identity and use Fubini theorem, we have:

$$(P_t V)(x) = V(x) + t(LV)(x) + \int_0^t (t-r)(P_r L^2 V)(x)\, dr.$$

Define $R_V(t;x,a) := \int_0^t (t-r)(P_r^a (L^a)^2 V)(x)\, dr$. Then

$$|R_V(t;x,a)| \le \|(L^a)^2 V\|_\infty \int_0^t (t-r)\, dr = \frac{t^2}{2} \|(L^a)^2 V\|_\infty,$$

which gives (70). □

Lemma E.5 (First-order bias for q_V^u: $\|q_V^u - q_V\|_\infty = \mathcal{O}(u)$)**.** *Under Assumptions E.2 and E.3, there exist $u_0 > 0$ and $C_q < \infty$ such that for all $u \in (0, u_0]$,*

$$\|q_V^u - q_V\|_\infty \le C_q\, u. \tag{71}$$

Proof. Fix (x,a). By Lemma E.4,

$$\mathbb{E}_x^a[V(X_u^a)] = V(x) + u(L^a V)(x) + \mathcal{O}(u^2),$$

uniformly in (x,a). Also $e^{-\beta u} = 1 - \beta u + \mathcal{O}(u^2)$, hence

$$\begin{aligned} e^{-\beta u}\mathbb{E}_x^a[V(X_u^a)] - V(x) &= \big(1 - \beta u + \mathcal{O}(u^2)\big)\big(V(x) + u(L^a V)(x) + \mathcal{O}(u^2)\big) - V(x) \\ &= u\big((L^a V)(x) - \beta V(x)\big) + \mathcal{O}(u^2), \end{aligned}$$

uniformly in (x,a). Divide by u and add $r(x,a)$, we obtain

$$q_V^u(x,a) = r(x,a) + (L^a V)(x) - \beta V(x) + \mathcal{O}(u) = q_V(x,a) + \mathcal{O}(u),$$

uniformly in (x,a), proving (71). □

Lemma E.6 (Richardson bias for $\tilde{q}_V^u$: $\|\tilde{q}_V^u - q_V\|_\infty = \mathcal{O}(u^2)$)**.** *Under Assumptions E.2 and E.3, there exist $u_0 > 0$ and $\tilde{C}_q < \infty$ such that for all $u \in (0, u_0]$,*

$$\|\tilde{q}_V^u - q_V\|_\infty \le \tilde{C}_q\, u^2. \tag{72}$$

Proof. Proof of Lemma E.5 implies the expansion of $q_V^u(x,a)$ contains only u and u^2 terms with no u^h, $h \in (1,2)$ in between:

$$q_V^u(x,a) = q_V(x,a) + u\, c_1(x,a) + u^2\, c_2(u;x,a),$$

where c_1 is bounded and c_2 is uniformly bounded for u small. Evaluating at $u/2$ and forming $\tilde{q}_V^u = 2q_V^{u/2} - q_V^u$ cancels the linear term:

$$\tilde{q}_V^u(x,a) = q_V(x,a) + u^2\Big(\tfrac{1}{2}c_2(u/2;x,a) - c_2(u;x,a)\Big),$$

which is $\mathcal{O}(u^2)$ uniformly in (x,a). □

E.4 HAMILTONIANS AND LIPSCHITZ LEMMAS

Recall the (soft) Hamiltonian for $\alpha > 0$,

$$H^{(\alpha)}(V)(x) := \alpha \log \int_{\mathcal{A}} \exp\Big(\frac{q_V(x,a)}{\alpha}\Big)\, da, \tag{73}$$

and its finite-horizon analogues on approximations q_V^u and $\tilde{q}_V^u$ of q-function:

$$H_u^{(\alpha)}(V)(x) := \alpha \log \int_{\mathcal{A}} \exp\Big(\frac{q_V^u(x,a)}{\alpha}\Big)\, da, \tag{74}$$

$$\tilde{H}_u^{(\alpha)}(V)(x) := \alpha \log \int_{\mathcal{A}} \exp\Big(\frac{\tilde{q}_V^u(x,a)}{\alpha}\Big)\, da. \tag{75}$$

Lemma E.7 (Lipschitz Hamiltonian bias)**.** *Fix $\alpha > 0$ and let $z, \bar{z} : \mathcal{A} \to \mathbb{R}$ be bounded. Then*

$$\Big|\alpha \log \int_{\mathcal{A}} e^{z(a)/\alpha}\, da - \alpha \log \int_{\mathcal{A}} e^{\bar{z}(a)/\alpha}\, da\Big| \le \|z - \bar{z}\|_\infty. \tag{76}$$

Consequently, for any bounded V,

$$\|H_u^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \le \|q_V^u - q_V\|_\infty, \tag{77}$$

$$\|\tilde{H}_u^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \le \|\tilde{q}_V^u - q_V\|_\infty. \tag{78}$$

Proof. For all a, $z(a) \le \bar{z}(a) + \|z - \bar{z}\|_\infty$, hence $e^{z(a)/\alpha} \le e^{\|z-\bar{z}\|_\infty/\alpha} e^{\bar{z}(a)/\alpha}$. Integrate and take logs to get

$$\log \int e^{z/\alpha} \le \frac{\|z - \bar{z}\|_\infty}{\alpha} + \log \int e^{\bar{z}/\alpha}.$$

Reverse the roles of $z, \bar{z}$ to obtain the other inequality, we can deduce (76). Apply this pointwise in x with $(z, \bar{z}) = (q_V^u(x,\cdot), q_V(x,\cdot))$ and $(z, \bar{z}) = (\tilde{q}_V^u(x,\cdot), q_V(x,\cdot))$, and then take the supremum over x, we obtain (77)–(78). □

Now we show that q_V^u is $\mathcal{O}(1/u)$-Lipschitz in V

Lemma E.8 (Lipschitz dependence of q_V^u on V)**.** *For any bounded functions V, W and any $u > 0$,*

$$\|q_V^u - q_W^u\|_\infty \le \frac{2}{u}\, \|V - W\|_\infty. \tag{79}$$

Consequently, for the Richardson rate,

$$\|\tilde{q}_V^u - \tilde{q}_W^u\|_\infty \le \frac{10}{u}\, \|V - W\|_\infty. \tag{80}$$

Proof. From (67), the $r(x,a)$ term cancels, so

$$q_V^u(x,a) - q_W^u(x,a) = \frac{e^{-\beta u}\mathbb{E}_x^a[(V-W)(X_u^a)] - (V-W)(x)}{u}.$$

Taking absolute values and using $|\mathbb{E}[(V-W)(X_u^a)]| \le \|V-W\|_\infty$ and $|(V-W)(x)| \le \|V-W\|_\infty$ gives (79). For Richardson, use $\tilde{q}_V^u = 2q_V^{u/2} - q_V^u$ and apply the first bound twice:

$$\|\tilde{q}_V^u - \tilde{q}_W^u\|_\infty \le 2\|q_V^{u/2} - q_W^{u/2}\|_\infty + \|q_V^u - q_W^u\|_\infty \le 2 \cdot \frac{2}{u/2}\|V-W\|_\infty + \frac{2}{u}\|V-W\|_\infty = \frac{10}{u}\|V-W\|_\infty.$$

□

E.5 One-step mismatch under stricter assumptions

Compared to Lemma D.4 (which gives an $\mathcal{O}(\tau^{3/2})$ remainder under standard assumptions), with stricter smooth assumptions, we can upgrade the local mismatch to $\mathcal{O}(\tau^2)$. This allows the τ^2 term in Theorem 4.2.

Lemma E.9 (Smooth local expansion of $\Phi_\tau^{(\alpha)}$)**.** *Under Assumptions E.2 and E.3, fix $\alpha \geq 0$ and $V \in C_b^4(\mathbb{R}^d)$. Then there exist $\tau_0 > 0$ and $C_\Phi < \infty$ such that for all $\tau \in (0, \tau_0]$,*

$$\Phi_\tau^{(\alpha)}(V) = V + \tau H^{(\alpha)}(V) + R_\tau^{(\alpha)}(V), \qquad \|R_\tau^{(\alpha)}(V)\|_\infty \leq C_\Phi\, \tau^2. \tag{81}$$

Equivalently,

$$\|\Phi_\tau^{(\alpha)}(V) - T_\tau^{(\alpha)}(V)\|_\infty \leq C_\Phi\, \tau^2. \tag{82}$$

Proof. We follow similar steps as in the proof of Lemma D.4. First we consider a fixed policy π, and recall from Lemma D.4 the policy-wise operator $\Phi_\tau^{\pi,(\alpha)}$:

$$(\Phi_\tau^{\pi,(\alpha)}V)(x) := \mathbb{E}_x^\pi\left[\int_0^\tau e^{-\beta t}\Big(r(X_t, a_t) - \alpha \log \pi(a_t \mid X_t)\Big)\, dt + e^{-\beta\tau}V(X_\tau)\right]. \tag{83}$$

We will show that, uniformly in x and uniformly over admissible π,

$$\Phi_\tau^{\pi,(\alpha)}V(x) = V(x) + \tau\, G_\alpha(V)(x, \pi) + \mathcal{O}(\tau^2), \tag{84}$$

where

$$G_\alpha(V)(x, \pi) := \int_{\mathcal{A}} q_V(x, a)\, \pi(da \mid x) - \alpha \int_{\mathcal{A}} \log \pi(a \mid x)\, \pi(da \mid x). \tag{85}$$

To see this, we again treat separately the terminal term and the running term.

Terminal term. Under Assumptions E.2 and C.2, the relaxed coefficients $(\tilde{b}(\cdot;\pi), \tilde{\sigma}(\cdot;\pi))$ are Lipschitz, and the relaxed generator L^π is well-defined and has bounded coefficients. Moreover $V \in C_b^4$ implies $L^\pi V$ and $(L^\pi)^2 V$ are bounded. Hence the same Dynkin-iteration argument as in Lemma E.4 (with L^a replaced by L^π) yields the second-order expansion

$$\mathbb{E}_x^\pi[V(X_\tau)] = V(x) + \tau(L^\pi V)(x) + \mathcal{O}(\tau^2), \tag{86}$$

uniformly in x and uniformly over admissible π. Multiplying by $e^{-\beta\tau} = 1 - \beta\tau + \mathcal{O}(\tau^2)$ gives

$$\mathbb{E}_x^\pi[e^{-\beta\tau}V(X_\tau)] = V(x) + \tau\big((L^\pi V)(x) - \beta V(x)\big) + \mathcal{O}(\tau^2). \tag{87}$$

Running term. Define the relaxed one-step reward-with-entropy

$$\tilde{r}^{(\alpha)}(x; \pi) := \int_{\mathcal{A}} \Big(r(x, a) - \alpha \log \pi(a \mid x)\Big)\, \pi(da \mid x).$$

Under Assumptions E.2 and C.2 and the policy regularity in Assumption C.3, the function $x \mapsto \tilde{r}^{(\alpha)}(x; \pi)$ is bounded and C^2 with bounded derivatives, uniformly over admissible π. Applying the same second-order semigroup expansion (again with generator L^π) to $\tilde{r}^{(\alpha)}(\cdot; \pi)$ implies

$$\mathbb{E}_x^\pi[\tilde{r}^{(\alpha)}(X_t; \pi)] = \tilde{r}^{(\alpha)}(x; \pi) + \mathcal{O}(t), \qquad t \in [0, 1], \tag{88}$$

uniformly in x and uniformly over π. Therefore,

$$\mathbb{E}_x^\pi\Big[\int_0^\tau e^{-\beta t}\tilde{r}^{(\alpha)}(X_t; \pi)\, dt\Big] = \int_0^\tau (1+\mathcal{O}(t))\big(\tilde{r}^{(\alpha)}(x; \pi)+\mathcal{O}(t)\big)\, dt = \tau\, \tilde{r}^{(\alpha)}(x; \pi)+\mathcal{O}(\tau^2) \tag{89}$$

uniformly in x and uniformly over π.

Combining (87) and (89) in (83) gives (84). Finally, note that the bracketed term in (84) equals

$$\tilde{r}^{(\alpha)}(x; \pi) + (L^\pi V)(x) - \beta V(x) = \int_{\mathcal{A}} q_V(x, a)\, \pi(da \mid x) - \alpha \int_{\mathcal{A}} \log \pi(a \mid x)\, \pi(da \mid x),$$

which is exactly $G_\alpha(V)(x, \pi)$ in (85).

Now we optimize over policies π to obtain the term $H^{(\alpha)}(V)$. By definition, $\Phi_\tau^{(\alpha)}(V)(x) = \sup_\pi \Phi_\tau^{\pi,(\alpha)}(V)(x)$. Hence, taking the supremum in (84) yields

$$\Phi_\tau^{(\alpha)}(V)(x) = V(x) + \tau \sup_\pi G_\alpha(V)(x,\pi) + \mathcal{O}(\tau^2).$$

Again, through a similar argument as in Lemma D.4 that invokes Lemma C.6, we have:

$$\sup_{\pi(\cdot|x)} \Big\{ \int q_V(x,a)\,\pi(da \mid x) - \alpha \int \log \pi(a \mid x)\,\pi(da \mid x) \Big\}$$
$$= \alpha \log \int_{\mathcal{A}} \exp\Big(\frac{q_V(x,a)}{\alpha}\Big)\,da = H^{(\alpha)}(V)(x)$$

For $\alpha = 0$ the same optimization reduces to $\sup_a q_V(x,a) = H^{(0)}(V)(x)$. Thus

$$\Phi_\tau^{(\alpha)}(V)(x) = V(x) + \tau H^{(\alpha)}(V)(x) + \mathcal{O}(\tau^2),$$

uniformly in x, which proves (81). The equivalent form (82) follows from the definition $T_\tau^{(\alpha)}(V) = V + \tau H^{(\alpha)}(V)$. □

Now define the finite-horizon Picard operator and its Richardson variant:

$$(T_{\tau,u}^{(\alpha)} V)(x) := V(x) + \tau\, H_u^{(\alpha)}(V)(x), \tag{90}$$
$$(T_{\tau,u}^{R,(\alpha)} V)(x) := V(x) + \tau\, \tilde{H}_u^{(\alpha)}(V)(x). \tag{91}$$

Here we have similar one-step mismatch as in Lemma D.4, but with better errors.

Lemma E.10 (One-step mismatch: finite-horizon and Richardson)**.** *Assume Assumptions E.2 and E.3, and fix $\alpha > 0$. Then there exist $\tau_0, u_0 > 0$ and $C < \infty$ such that for all $\tau \in (0, \tau_0]$, $u \in (0, u_0]$, and all $V \in C_b^4(\mathbb{R}^d)$,*

$$\|\Phi_\tau^{(\alpha)}(V) - T_{\tau,u}^{(\alpha)}(V)\|_\infty \le C\big(\tau^2 + \tau u\big), \tag{92}$$
$$\|\Phi_\tau^{(\alpha)}(V) - T_{\tau,u}^{R,(\alpha)}(V)\|_\infty \le C\big(\tau^2 + \tau u^2\big). \tag{93}$$

Proof. By Lemma E.9,

$$\|\Phi_\tau^{(\alpha)}(V) - T_\tau^{(\alpha)}(V)\|_\infty \le C_\Phi \tau^2.$$

For the finite-horizon Picard operator,

$$\|T_\tau^{(\alpha)}(V) - T_{\tau,u}^{(\alpha)}(V)\|_\infty = \tau\, \|H^{(\alpha)}(V) - H_u^{(\alpha)}(V)\|_\infty \le \tau\, \|q_V - q_V^u\|_\infty,$$

using Lemma E.7. Then Lemma E.5 yields $\|q_V - q_V^u\|_\infty \le C_q u$, hence

$$\|\Phi_\tau^{(\alpha)}(V) - T_{\tau,u}^{(\alpha)}(V)\|_\infty \le C_\Phi \tau^2 + \tau C_q u \le C(\tau^2 + \tau u).$$

The Richardson bound's proof is identical through using the following inequality from Lemmas E.6 and E.7:

$$\|H^{(\alpha)}(V) - \tilde{H}_u^{(\alpha)}(V)\|_\infty \le \|\tilde{q}_V^u - q_V\|_\infty \le \tilde{C}_q u^2$$

□

E.6 Proof of Theorem 4.2

Theorem 4.2 (Convergence with q-function)**.** *Under Assumptions E.2 and E.3, as $k \to \infty$, the Richardson iteration $V_{k+1} = V_k + \tau \tilde{H}_u^{(\alpha)}(V_k)$ satisfies $\|V_k - V^{(\alpha)}\|_\infty = \mathcal{O}(\tau + u^2)$ and $\|\tilde{q}_{V_k}^u - q_{V^{(\alpha)}}\|_\infty = \mathcal{O}(\tau/u + u)$. In addition, the first-order iteration $V_{k+1} = V_k + \tau H_u^{(\alpha)}(V_k)$ satisfies $\|V_k - V^{(\alpha)}\|_\infty = \mathcal{O}(\tau + u)$ and $\|q_{V_k}^u - q_{V^{(\alpha)}}\|_\infty = \mathcal{O}\big(u^{-1/2}(\tau + u)^{3/4} + \tau + u\big)$.*

Proof. Let $V^* = V^{(\alpha)}$. Denote the Richardson and first-order iterates by $V_k^{(R)}$ and $V_k^{(1)}$, respectively.

Step 1: Value convergence. By the contraction property in Lemma D.1,

$$\|\Phi_\tau^{(\alpha)}(V) - \Phi_\tau^{(\alpha)}(V^*)\|_\infty \le e^{-\beta\tau}\|V - V^*\|_\infty.$$

For the Richardson iteration, combining this contraction with (93) gives:

$$\|V_{k+1}^{(R)} - V^*\|_\infty \le e^{-\beta\tau}\|V_k^{(R)} - V^*\|_\infty + C(\tau^2 + \tau u^2).$$

Similarly, for the first-order iteration, (92) gives:

$$\|V_{k+1}^{(1)} - V^*\|_\infty \le e^{-\beta\tau}\|V_k^{(1)} - V^*\|_\infty + C(\tau^2 + \tau u).$$

Unrolling these recursions yields:

$$\|V_k^{(R)} - V^*\|_\infty \le e^{-\beta\tau k}\|V_0 - V^*\|_\infty + \frac{C(\tau^2 + \tau u^2)}{1 - e^{-\beta\tau}},$$

and

$$\|V_k^{(1)} - V^*\|_\infty \le e^{-\beta\tau k}\|V_0 - V^*\|_\infty + \frac{C(\tau^2 + \tau u)}{1 - e^{-\beta\tau}}.$$

Since $1 - e^{-\beta\tau} \asymp \tau$, after the exponentially decaying transient term vanishes as $k \to \infty$,

$$\|V_k^{(R)} - V^*\|_\infty = \mathcal{O}(\tau + u^2), \qquad \|V_k^{(1)} - V^*\|_\infty = \mathcal{O}(\tau + u).$$

Step 2: Richardson q-convergence. By the triangle inequality,

$$\|\tilde{q}^u_{V_k^{(R)}} - q_{V^*}\|_\infty \le \|\tilde{q}^u_{V_k^{(R)}} - \tilde{q}^u_{V^*}\|_\infty + \|\tilde{q}^u_{V^*} - q_{V^*}\|_\infty.$$

Using Lemmas E.6 and E.8,

$$\|\tilde{q}^u_{V_k^{(R)}} - q_{V^*}\|_\infty \le \frac{10}{u}\|V_k^{(R)} - V^*\|_\infty + \tilde{C}_q u^2.$$

Therefore, after the transient term vanishes,

$$\|\tilde{q}^u_{V_k^{(R)}} - q_{V^*}\|_\infty = \mathcal{O}\Big(\frac{\tau}{u} + u\Big).$$

Hence, if $\tau, u \to 0$ with $\tau/u \to 0$, then $V_k^{(R)} \to V^*$ and $\tilde{q}^u_{V_k^{(R)}} \to q_{V^*}$.

Step 3: First-order q-convergence. Let $e_k := V_k^{(1)} - V^*$. By assumption E.3 and through applying the Landau–Kolmogorov inequality [14, Chapter 1] along each unit direction, we have:

$$\mathrm{Lip}(e_k) \le C_M \|e_k\|_\infty^{3/4}.$$

From the definition of q_V^u,

$$q^u_{V_k^{(1)}}(x,a) - q^u_{V^*}(x,a) = e^{-\beta u}\frac{\mathbb{E}_x^a[e_k(X_u^a)] - e_k(x)}{u} + \frac{e^{-\beta u} - 1}{u}e_k(x).$$

By Lemma C.5,

$$|\mathbb{E}_x^a[e_k(X_u^a)] - e_k(x)| \le \mathrm{Lip}(e_k)\mathbb{E}_x^a\|X_u^a - x\| \le C\,\mathrm{Lip}(e_k)\sqrt{u}.$$

Since $(1 - e^{-\beta u})/u \le \beta$,

$$\|q^u_{V_k^{(1)}} - q^u_{V^*}\|_\infty \le C_M u^{-1/2}\|V_k^{(1)} - V^*\|_\infty^{3/4} + \beta\|V_k^{(1)} - V^*\|_\infty.$$

Combining this estimate with the first-order modelling bias in Lemma E.5 gives

$$\|q^u_{V_k^{(1)}} - q_{V^*}\|_\infty \le C_M u^{-1/2}\|V_k^{(1)} - V^*\|_\infty^{3/4} + \beta\|V_k^{(1)} - V^*\|_\infty + C_q u.$$

Substituting the value estimate from Step 1, after the transient vanishes,

$$\|q^u_{V_k^{(1)}} - q_{V^*}\|_\infty = \mathcal{O}\Big(u^{-1/2}(\tau + u)^{3/4} + \tau + u\Big).$$

If $\tau, u \to 0$ with $\tau/u \to 0$, then $\tau + u = \mathcal{O}(u)$, so the right-hand side is $\mathcal{O}(u^{1/4}) + \mathcal{O}(u) \to 0$. Thus $V_k^{(1)} \to V^*$ and $q^u_{V_k^{(1)}} \to q_{V^*}$ as well. The two cases together prove the result. □

F PROOF OF CONVERGENCE FOR CT-SAC (THEOREM 4.3 AND COROLLARY 4.4)

F.1 FROM (V_k, q_k) TO A SINGLE Q_k

We prove that the single-critic update is an algebraic re-expression of the decoupled (V_k, q_k) iteration, hence inherits its convergence guarantees. Fix an algorithmic step size $\tau > 0$ and a holding time $u > 0$ with $\gamma := e^{-\beta u}$. Recall that

$$q_V^u(x,a) := \frac{\gamma\, \mathbb{E}_x^a[V(X_u)] - V(x)}{u} + r(x,a). \tag{94}$$

We will also use the shorthand

$$\Delta_u V(x,a) := \gamma\, \mathbb{E}_x^a[V(X_u)] - V(x), \qquad q_V^u(x,a) = \frac{\Delta_u V(x,a)}{u} + r(x,a). \tag{95}$$

Lemma F.1 (One-step Q-update induced by (V, q))**.** *Assume that at iteration k we have a pair (V_k, q_k) and a policy π_k such that*

$$V_{k+1}(x) = V_k(x) + \tau \mathcal{Q}^{(\alpha)}(q_k)(x) \quad \textit{(Hamiltonian flow update)}, \tag{96}$$

and the finite-horizon approximation q_{k+1}:

$$q_{k+1}(x,a) = \frac{\gamma\, \mathbb{E}_x^a[V_{k+1}(X_u)] - V_{k+1}(x)}{u} + r(x,a). \tag{97}$$

Define the single critic $Q_k(x,a) := V_k(x) + q_k(x,a)$. Then $Q_{k+1} := V_{k+1} + q_{k+1}$ satisfies the closed-form update

$$\begin{aligned} Q_{k+1}(x,a) = (1-\tau)Q_k(x,a) + \tau r(x,a) + \tau\, \mathbb{E}_{a\sim\pi_k(\cdot|x)}\big[\tilde{Q}_k(x,a)\big] \\ + \tau\, \frac{\gamma\, \mathbb{E}_{a'\sim\pi_k(\cdot|X_u)}\big[\tilde{Q}_k(X_u,a')\big] - \mathbb{E}_{a\sim\pi_k(\cdot|x)}\big[\tilde{Q}_k(x,a)\big]}{u}, \end{aligned} \tag{98}$$

where $\tilde{Q}_k(x,a)$ is defined as $Q_k(x,a) - \alpha \log \pi_k(a \mid x)$ with $\pi_k(a \mid x) \sim \exp(q_k(x,a)/\alpha) \sim \exp(Q_k(x,a)/\alpha)$.

Note that this is the same equation as Equation (28)*. We state the equation again here for convenience reference.*

Proof. From Lemma C.6, we immediately have $V_{k+1}(x) = V_k(x) + \tau\, \mathbb{E}_{a\sim\pi_k(\cdot|x)}\Big[q_k(x,a) - \alpha \log \pi_k(a \mid x)\Big]$, for $\pi_k(.|x) \sim \exp(q_k(x,.)/\alpha)$. Using $q_k = Q_k - V_k$ gives:

$$\begin{aligned} V_{k+1}(x) &= V_k(x) + \tau\, \mathbb{E}_{a\sim\pi_k}\Big[Q_k(x,a) - V_k(x) - \alpha \log \pi_k(a \mid x)\Big] \\ &= (1-\tau)V_k(x) + \tau\, \mathbb{E}_{a\sim\pi_k(\cdot|x)}\big[\tilde{Q}_k(x,a)\big]. \end{aligned} \tag{99}$$

Plugging Equation (99) into Equation (97) gives

$$\begin{aligned} q_{k+1}(x,a) &= \frac{1}{u}\Big(\gamma\, \mathbb{E}_x^a[V_{k+1}(X_u)] - V_{k+1}(x)\Big) + r(x,a) \\ &= \frac{1}{u}\Big(\gamma(1-\tau)\mathbb{E}_x^a[V_k(X_u)] + \gamma\tau\, \mathbb{E}_x^a\Big[\mathbb{E}_{a'}[\tilde{Q}_k(X_u,a')]\Big] \\ &\qquad - (1-\tau)V_k(x) - \tau\, \mathbb{E}_a[\tilde{Q}_k(x,a)]\Big) + r(x,a). \end{aligned} \tag{100}$$

Rearranging the terms, we get (we drop $\mathbb{E}_x^a$ for the last term for readability):

$$\begin{aligned} q_{k+1}(x,a) &= (1-\tau)\frac{\gamma \mathbb{E}_x^a[V_k(X_u)] - V_k(x)}{u} + r(x,a) + \frac{\tau}{u}\Big(\gamma\, \mathbb{E}_{a'}[\tilde{Q}_k(X_u,a')] - \mathbb{E}_a[\tilde{Q}_k(x,a)]\Big) \\ &= (1-\tau)\big(q_k(x,a) - r(x,a)\big) + r(x,a) + \frac{\tau}{u}\Big(\gamma\, \mathbb{E}_{a'}[\tilde{Q}_k(X_u,a')] - \mathbb{E}_a[\tilde{Q}_k(x,a)]\Big). \end{aligned} \tag{101}$$

Algorithm 2 Continuous-time TD3 (CT-TD3)

Input: discount rate β, Euler step $\tau > 0$, actor delay $d \in \mathbb{N}$, exploration std σ_{expl}, target smoothing std σ_{targ}, clip $c > 0$.
Initialize twin critics $Q_{\theta_1}, Q_{\theta_2}$ and deterministic actor μ_ϕ.
Initialize target networks $Q_{\bar{\theta}_1} \leftarrow Q_{\theta_1}$, $Q_{\bar{\theta}_2} \leftarrow Q_{\theta_2}$, $\mu_{\bar{\phi}} \leftarrow \mu_\phi$.
for $k = 0, 1, 2, \ldots$ **do**
  Collect transitions (x, a, r, x', u), where $x = X_t$, $x' = X_{t+u}$, and u can vary across samples. Use behavior action $a = \mu_\phi(x) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2_{\text{expl}} I)$ (clip to action bounds if needed). Store in an off-policy replay buffer.
  Sample a mini-batch $\mathcal{B}$ of (x, a, r, x', u) from the replay buffer and set $\gamma = e^{-\beta u}$ per sample.
  Target policy smoothing:

$$\tilde{a}' = \mu_{\bar{\phi}}(x') + \epsilon', \qquad \epsilon' \sim \text{clip}\big(\mathcal{N}(0, \sigma^2_{\text{targ}} I), -c, c\big),$$

  Compute clipped double-Q bootstrap:

$$\bar{Q}_{\bar{\theta}}(x', \tilde{a}') := \min\{Q_{\bar{\theta}_1}(x', \tilde{a}'),\, Q_{\bar{\theta}_2}(x', \tilde{a}')\}.$$

  Critic updates: for $i \in \{1, 2\}$, form the target

$$y_i := (1-\tau) Q_{\bar{\theta}_i}(x, a) + \tau\, \bar{Q}_{\bar{\theta}}\big(x, \mu_{\bar{\phi}}(x)\big) + \tau\, r + \tau\, \frac{\gamma\, \bar{Q}_{\bar{\theta}}(x', \tilde{a}') - \bar{Q}_{\bar{\theta}}\big(x, \mu_{\bar{\phi}}(x)\big)}{u},$$

  and update θ_i by regression on $\mathcal{B}$:

$$\theta_i \leftarrow \arg\min_\theta \; \mathbb{E}_{(x,a,r,x',u)\sim\mathcal{B}} \big(Q_\theta(x, a) - y_i\big)^2.$$

  if $k \bmod d = 0$ **then**
    Delayed policy update: update actor using critic Q_{θ_1}:

$$\phi \leftarrow \phi + \eta_\pi\, \mathbb{E}_{x\sim\mathcal{B}} \Big[\nabla_\phi \mu_\phi(x)\, \nabla_a Q_{\theta_1}(x, a)\big|_{a=\mu_\phi(x)}\Big].$$

  end if
end for

Finally, add $V_{k+1}(x)$ from Equation (101) to obtain

$$\begin{aligned} Q_{k+1}(x,a) &= V_{k+1}(x) + q_{k+1}(x,a) \\ &= (1-\tau)V_k(x) + \tau \mathbb{E}_a[\tilde{Q}_k(x,a)] + (1-\tau) q_k(x,a) + \tau r(x,a) \\ &\quad + \frac{\tau}{u}\Big(\gamma\, \mathbb{E}_{a'}[\tilde{Q}_k(X_u, a')] - \mathbb{E}_a[\tilde{Q}_k(x,a)]\Big) \\ &= (1-\tau) Q_k(x,a) + \tau r(x,a) + \tau \mathbb{E}_a[\tilde{Q}_k(x,a)] + \tau \frac{\gamma \mathbb{E}_{a'}[\tilde{Q}_k(X_u,a')] - \mathbb{E}_a[\tilde{Q}_k(x,a)]}{u}. \end{aligned} \tag{102}$$

This is exactly Equation (98) (or Equation (28)). □

Note. For implementation, it is convenient to introduce a "fast" update

$$Q^{\text{fast}}_{k+1}(s,a) = \frac{Q_{k+1}(s,a) - (1-\tau) Q_k(s,a)}{\tau} \tag{103}$$

The full update can then be written as:

$$Q^{\text{fast}}_{k+1}(s,a) = r(s,a) + \mathbb{E}_a\big[\tilde{Q}_k(s,a)\big] + \frac{\gamma \mathbb{E}_{a'}\big[\tilde{Q}_k(s',a')\big] - \mathbb{E}_a\big[\tilde{Q}_k(s,a)\big]}{u} \tag{104}$$

$$Q_{k+1}(s,a) = \tau Q^{\text{fast}}_{k+1}(s,a) + (1-\tau) Q_k(s,a) \tag{105}$$

When $u = 1$, this reduces to a Q-update closely aligned with the discrete-time counterpart. Below is a deterministic version (CT-TD3) of CT-SAC (See Algorithm 2).

Richardson update. We now state the Richardson version of Q-function update. Fix a holding time $u > 0$ and define $\gamma := e^{-\beta u}$ and $\gamma_{1/2} := e^{-\beta u/2}$. Let $X_{u/2}$ denote the state reached after time $u/2$ when starting from (x, a). Define the policy-averaged soft value functional

$$S_k(x) := \mathbb{E}_{a\sim\pi_k(\cdot|x)}\big[\tilde{Q}^{\mathrm{R}}_k(x,a)\big], \qquad \tilde{Q}^{\mathrm{R}}_k(x,a) := Q^{\mathrm{R}}_k(x,a) - \alpha \log \pi_k(a \mid x), \tag{106}$$

and the Richardson critic decomposition

$$Q^{\mathrm{R}}_k(x,a) := V_k(x) + \tilde{q}^u_{V_k}(x,a), \tag{107}$$

Then the induced single-critic update of Q^{R}_k has the form:

$$\begin{aligned} Q^{\mathrm{R}}_{k+1}(x,a) = (1-\tau)Q^{\mathrm{R}}_k(x,a) + \tau r(x,a) + \tau S_k(x) \\ + \frac{\tau}{u}\Big(4\gamma_{1/2}\,\mathbb{E}^a_x\big[S_k(X_{u/2})\big] - \gamma\,\mathbb{E}^a_x\big[S_k(X_u)\big] - 3S_k(x)\Big). \end{aligned} \tag{108}$$

Detailed derivations are given in Lemma I.2. We now continue with the algorithm convergence.

Lemma F.2 (Inductive preservation of $Q_k = V_k + q_k$)**.** *Assume (V_0, q_0) satisfy the finite-horizon relation $q_0 = q^u_{V_0}$ from Equation* (94) *and define $Q_0 := V_0 + q_0$. Let (V_k, q_k) be generated by Equation* (96)*–Equation* (97) *and let Q_k be generated by Equation* (98)*. Then for all $k \geq 0$,*

$$Q_k(x,a) = V_k(x) + q_k(x,a). \tag{109}$$

Proof. By construction, Equation (109) holds for $k = 0$. Assume it holds for some k. By Lemma F.1, the update Equation (98) produced from (V_k, q_k) satisfies $Q_{k+1} = V_{k+1} + q_{k+1}$. This completes the induction. □

We now show that the single critic converges by combining the value and rate convergence bounds.

Theorem 4.3 (Theoretical Algorithm Convergence)**.** *Consider the exact single-critic iteration Q_k, for either Equation* (28) *or its Richardson-corrected variant, with no optimization or statistical error. Then as $k \to \infty$, Q_k converges to the optimal value $Q^{(\alpha)} = V^{(\alpha)} + q_{V^{(\alpha)}}$.*

Proof. From Lemma F.2, we get:

$$Q_k - Q^{(\alpha)} = (V_k - V^{(\alpha)}) + (q_k - q_{V^{(\alpha)}}), \tag{110}$$

Therefore, by the triangle inequality,

$$\|Q_k - Q^{(\alpha)}\|_\infty \leq \|V_k - V^{(\alpha)}\|_\infty + \|q_k - q_{V^{(\alpha)}}\|_\infty. \tag{111}$$

For the first-order single-critic update Equation (28), Lemma F.2 gives $Q_k = V_k + q^u_{V_k}$. For the Richardson-corrected update, the analogous Richardson algebra gives $Q^{\mathrm{R}}_k = V_k + \tilde{q}^u_{V_k}$. In both cases, the corresponding value and q-function terms converge by Theorem 4.2. Hence in either case,

$$\|Q_k - Q^{(\alpha)}\|_\infty \to 0$$

under the corresponding joint refinement. □

We now sketch a standard finite-sample extension that converts the exact convergence into a statistical bound for the learned critic $\hat{Q}_k$ produced by neural network fitting. We first state standard i.i.d. generalization bounds that we use to control the statistical error of the critic regression step.

F.2 Learning-theoretic ingredients (i.i.d. regression)

Throughout this section, $\mathcal{X}$ denotes an input space, $\mathcal{Y} \subset \mathbb{R}$ a bounded label space, and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Let $\mathcal{H}$ be a hypothesis class of functions $h : \mathcal{X} \to \mathcal{Y}$ and let $l(h, (x, y)) = \phi(h(x), y)$ be a loss with $\phi(\cdot, y)$ γ-Lipschitz for all y. For i.i.d. samples $Z_i = (X_i, Y_i) \sim \mathbb{P}$, define the population and empirical risks

$$e(h) := \mathbb{E}[l(h, Z)], \qquad E_n(h) := \frac{1}{n}\sum_{i=1}^n l(h, Z_i). \tag{112}$$

Definition F.3 (Rademacher complexity)**.** For $\mathcal{T} \subseteq \mathbb{R}^n$, its Rademacher complexity is

$$\mathrm{Rad}(\mathcal{T}) := \mathbb{E}\Big[\sup_{t\in\mathcal{T}} \frac{1}{n}\sum_{i=1}^{n} B_i t_i\Big], \tag{113}$$

where B_i are i.i.d. Rademacher random variables.

Besides Rademacher complexity, we also define the following sets: $\mathcal{H} \circ \{Z_1, \cdots, Z_n\} := \{(h(X_1), \cdots, h(X_n)),\ h \in \mathcal{H}\} \subseteq \mathbb{R}^n$, and $\mathcal{L} \circ \{Z_1, \cdots, Z_n\} := \{(l(h, Z_1), \cdots, l(h, Z_n)),\ h \in \mathcal{H}\} \subseteq \mathbb{R}^n$. We state the following standard PAC-type results [17]:

Lemma F.4 (Lipschitz contraction for losses)**.** *If $\phi(\cdot, y)$ is γ-Lipschitz for all y, then*

$$\mathbb{E}\Big[\sup_{h\in\mathcal{H}}(e(h)-E_n(h))\Big] \le 2\,\mathbb{E}\big[\mathrm{Rad}(\mathcal{L}\circ\{Z_1,\ldots,Z_n\})\big] \le 2\gamma\,\mathbb{E}\big[\mathrm{Rad}(\mathcal{H}\circ\{Z_1,\ldots,Z_n\})\big]. \tag{114}$$

Lemma F.5 (High-probability uniform bound)**.** *Assume $l(h, Z) \in [0, c]$ almost surely. Then with probability at least $1 - \delta$,*

$$\sup_{h\in\mathcal{H}}\big(e(h) - E_n(h)\big) \le 4\,\mathrm{Rad}(\mathcal{L} \circ \{Z_1, \ldots, Z_n\}) + c\sqrt{\frac{2\log(1/\delta)}{n}}. \tag{115}$$

Lemma F.6 (Rademacher complexity for bounded neural nets)**.** *For a hypothesis class $\mathcal{H}$ consisting of (regularized) neural networks with bounded weights, there exist constants $C_1, C_2 > 0$ such that for n i.i.d. samples,*

$$\mathrm{Rad}(\mathcal{H} \circ \{Z_1, \ldots, Z_n\}) \le \frac{1}{\sqrt{n}}\big(C_1 + C_2\sqrt{\log d}\big), \tag{116}$$

where d is an ambient dimension parameter of the network class.

F.3 Proof of Corollary 4.4

Definition F.7. With $\tilde{Q}(x, a) := Q(x, a) - \alpha \log \pi(a \mid x)$, and $\pi(a|x) \sim \exp(Q(x, a)/\alpha)$, we define the single-critic update operator arising from Equation (28):

$$\begin{aligned}\big(\mathcal{F}_{\tau,u}(Q)\big)(x,a) := (1-\tau)Q(x,a) + \tau r(x,a) + \tau\mathbb{E}_{a\sim\pi(\cdot|x)}[\tilde{Q}(x,a)] \\ + \tau\frac{\gamma\,\mathbb{E}_{a'\sim\pi(\cdot|x')}[\tilde{Q}(x',a')] - \mathbb{E}_{a\sim\pi(\cdot|x)}[\tilde{Q}(x,a)]}{u}\end{aligned} \tag{117}$$

Definition F.8. With Richardson q-function $\tilde{q}_V^u := 2q_V^{u/2} - q_V^u$, the single-critic update operator is:

$$\begin{aligned}&\big(\mathcal{F}^{\mathrm{Rich}}_{\tau,u}(Q)\big)(x,a) := (1-\tau)Q(x,a) + \tau r(x,a) + \tau\mathbb{E}_{a\sim\pi(\cdot|x)}[\tilde{Q}(x,a)] \\ &\quad + \frac{\tau}{u}\Big(4e^{-\beta u/2}\mathbb{E}_{a'\sim\pi(\cdot|x'_{u/2})}\big[\tilde{Q}(x'_{u/2},a')\big] - e^{-\beta u}\mathbb{E}_{a''\sim\pi(\cdot|x'_u)}\big[\tilde{Q}(x'_u,a'')\big] - 3\mathbb{E}_{a\sim\pi(\cdot|x)}[\tilde{Q}(x,a)]\Big)\end{aligned} \tag{118}$$

where $x'_{u/2} = X_{t+u/2}$ and $x'_u = X_{t+u}$.

Approximation procedure. At each iteration k, we draw n_k i.i.d. transition samples

$$(X_i, A_i, R_i, X'_i, U_i)_{i=1}^{n_k}, \tag{119}$$

from a replay distribution (treated as i.i.d. for analysis), and form a bounded regression target $Y_k(x, a, r, x', u)$ corresponding to the right-hand side of the population update Equation (28). We then fit $\hat{Q}_{k+1} \in \mathcal{H}_Q$ by (approximately) minimizing the empirical loss $\frac{1}{n_k}\sum_{i=1}^{n_k}\phi(\hat{Q}_{k+1}(X_i, A_i), Y_k(Z_i))$. Assume the output range of $\mathcal{H}_Q$ and the targets are uniformly bounded. Let ν denote the state–action sampling distribution induced by the data collection scheme (e.g. replay buffer sampling), and define the $L^2(\nu)$ norm by:

$$\|f\|_{L^2(\nu)} := \Big(\mathbb{E}_{(X,A)\sim\nu}[\,|f(X,A)|^2\,]\Big)^{1/2}.$$

Let Q_k denote the exact iterate and $\hat{Q}_{k+1}$ the learned iterate obtained by regression at iteration $k+1$. At each step k, the target Y_k is exactly the value $\big(\mathcal{F}_{\tau,u}(Q)\big)$ on realized samples. As a result, thanks to Lemmas F.4 to F.6, we can estimate the one-step statistical error (which is a random variable):

$$\mathrm{stat}_k := \|\hat{Q}_{k+1} - \mathcal{F}_{\tau,u}(\hat{Q}_k)\|_{L^2(\nu)}.$$

In this case, we have:

$$\mathrm{stat}_k \lesssim \frac{1}{\sqrt{n_k}}\left(1 + \sqrt{\log d} + \sqrt{\log(1/\delta)}\right) \qquad \text{with probability} \geq 1-\delta, \tag{120}$$

Note that the replay buffer sampling distribution $\nu = \nu_k$ can change at each iteration. However, for large enough n_k, we can easily transfer from $L^2(\nu)$ to $L^2(\nu_k)$ given that ν_k is regular enough. Hence, for simplicity, we proceed with a fixed replay buffer sampling ν.

Lemma F.9 (Lipschitz of critic mapping)**.** *Assume the policy $\pi_k(\cdot \mid x)$ used inside $\tilde{Q}_k(x,a) := Q_k(x,a) - \alpha \log \pi_k(a \mid x)$ is treated as fixed when comparing two critics (policy-evaluation step). Then the mapping $\mathcal{F}_{\tau,u}$ is Lipschitz in $L^2(\nu)$:*

$$\|\mathcal{F}_{\tau,u}(Q) - \mathcal{F}_{\tau,u}(Q')\|_{L^2(\nu)} \leq \Big(1 + \frac{\tau}{u}\Big)\|Q - Q'\|_{L^2(\nu)}. \tag{121}$$

Proof. The update $\mathcal{F}_{\tau,u}$ is affine in Q through terms of the form $Q(x,a)$, $\mathbb{E}_{a\sim\pi(\cdot|x)}[Q(x,a)]$, and $\mathbb{E}_{a'\sim\pi(\cdot|x')}[Q(x',a')]$, all multiplied by coefficients bounded by $1+\tau/u$. Using Jensen's inequality for conditional expectations $\big\|\mathbb{E}[Q(X,\cdot) \mid X]\big\|_{L^2} \leq \|Q(X,\cdot)\|_{L^2}$ yields the stated Lipschitz bound. □

Corollary 4.4 (Algorithm Convergence)**.** *Suppose at each iteration k, n_k samples are used in the gradient update. Then, for any pair of $(\epsilon,\delta) > 0$, there exists an iteration L together with the sequence $n_1, n_2, \cdots, n_L$ so that with the probability of at least $1-\delta$, $\mathbb{E}\left[|Q_L(X,A) - Q^{(\alpha)}(X,A)|^2\right] < \epsilon$ for random state-action pair (X,A). Here $Q^{(\alpha)} = V^{(\alpha)} + q_{V^{(\alpha)}}$ is the optimal function, and Q_L is the learned function from Algorithm 1.*

Proof. Given an $\epsilon > 0$, from Theorem 4.3, since the algorithmic error go to 0, we can choose an iteration L so that $\|Q_L - Q^{(\alpha)}\|_{L^2(\nu)} \leq \|Q_L - Q^{(\alpha)}\|_\infty < \epsilon/2$. Additionally, choose n_j large enough, so that $\mathrm{stat}_{\mathrm{j}}^{(2)} < (1/L_0^{L-1-j})\dfrac{\epsilon}{2L}$ for probability of at least $1-\delta/L$ for $L_0 = 1+\tau/u$.

Now let Q_k be the exact iteration results and $\hat{Q}_k$ the learned one. Define the statistical deviation in $L^2(\nu)$:

$$\epsilon_k^{(2)} := \|\hat{Q}_k - Q_k\|_{L^2(\nu)}.$$

Then we have the decomposition,

$$\begin{aligned}\epsilon_{k+1}^{(2)} &= \|\hat{Q}_{k+1} - Q_{k+1}\|_{L^2(\nu)} \\ &\leq \|\hat{Q}_{k+1} - \mathcal{F}_{\tau,u}(\hat{Q}_k)\|_{L^2(\nu)} + \|\mathcal{F}_{\tau,u}(\hat{Q}_k) - \mathcal{F}_{\tau,u}(Q_k)\|_{L^2(\nu)}.\end{aligned} \tag{122}$$

The first term is the one-step regression error stat_k. The second term is bounded by Lemma F.9:

$$\|\mathcal{F}_{\tau,u}(\hat{Q}_k) - \mathcal{F}_{\tau,u}(Q_k)\|_{L^2(\nu)} \leq \Big(1 + \frac{\tau}{u}\Big)\epsilon_k^{(2)}.$$

Hence,

$$\epsilon_{k+1}^{(2)} \leq \mathrm{stat}_k + \Big(1 + \frac{\tau}{u}\Big)\epsilon_k^{(2)}. \tag{123}$$

Given the same initial point with $\epsilon_0^{(2)} = 0$, an inductive argument gives:

$$\epsilon_L^{(2)} \leq \sum_{j=0}^{L-1}\Big(1 + \frac{\tau}{u}\Big)^{L-1-j}\mathrm{stat}_j = \sum_{j=0}^{L-1} L_0^{L-1-j}\mathrm{stat}_j \tag{124}$$

Based on the choice of n_k, the RHS can further be bounded by $\sum_{j=0}^{L-1}\epsilon/(2L) = \epsilon/2$ with probability of at least $1 - L(\delta/L) = 1-\delta$. Finally, with probability of at least $1-\delta$, we have the decomposition:

$$\|\hat{Q}_L - Q^{(\alpha)}\|_{L^2(\nu)} \leq \underbrace{\|Q_L - Q^{(\alpha)}\|_{L^2(\nu)}}_{\text{algorithmic error}} + \underbrace{\|\hat{Q}_L - Q_L\|_{L^2(\nu)}}_{=\epsilon_L^{(2)}\ \text{statistical error}} < \epsilon/2 + \epsilon/2 = \epsilon \tag{125}$$

□

We defer the theoretical analysis with more refined control of sample complexity and regret bound to Section H.

G RANDOM TIME STEP U

Many environments produce irregular times U, and here we provide similar convergence results for (random time) U. The deterministic finite-horizon approximation results extend directly by conditioning on U and averaging. The key observation is that any deterministic Hamiltonian bias bound in u transfers to random U by replacing u^γ with $\mathbb{E}[U^\gamma]$.

Lemma G.1 (Hamiltonian bias)**.** *Fix $\alpha \geq 0$ and a bounded test function V. Let U be a random variable taking values in $(0, 1]$, independent of the state and control. Assume that for some $\gamma > 0$ and constant $C_H < \infty$,*

$$\|H_u^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \leq C_H\, u^\gamma, \qquad \forall u \in (0, 1]. \tag{126}$$

Define the random Hamiltonian $H_U^{(\alpha)}(V)$ by evaluation at $u = U$, and its averaged version $\bar{H}_U^{(\alpha)}(V) := \mathbb{E}[H_U^{(\alpha)}(V)]$ (expectation over U only). Then,

$$\|\bar{H}_U^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \leq C_H\, \mathbb{E}[U^\gamma]. \tag{127}$$

The same statement holds if $H_u^{(\alpha)}$ is replaced by the Richardson Hamiltonian $\tilde{H}_u^{(\alpha)}$.

Proof. By Jensen and the deterministic bound (126),

$$\begin{aligned}\|\bar{H}_U^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty &= \Big\|\mathbb{E}\big[H_U^{(\alpha)}(V) - H^{(\alpha)}(V)\big]\Big\|_\infty \\ &\leq \mathbb{E}\big[\|H_U^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty\big] \leq C_H\, \mathbb{E}[U^\gamma]\end{aligned}$$

The Richardson case is identical. □

Theorem G.2 (Picard iteration with random time step)**.** *Fix $\alpha \geq 0$ and $\tau > 0$. Let $\Phi_\tau^{(\alpha)}$ be the dynamic-programming semigroup from* (49)*. Consider the averaged Picard operator*

$$(\widetilde{T}_\tau^{(\alpha)}V)(x) := V(x) + \tau\, \bar{H}_U^{(\alpha)}(V)(x) = V(x) + \tau\, \mathbb{E}\big[H_U^{(\alpha)}(V)(x)\big], \tag{128}$$

and the iterates $\widetilde{V}_{k+1} := \widetilde{T}_\tau^{(\alpha)}(\widetilde{V}_k)$.

Assume further that the local expansion error satisfies, for all $V \in \mathcal{D}$,

$$\|\Phi_\tau^{(\alpha)}(V) - (V + \tau H^{(\alpha)}(V))\|_\infty \leq C_{\mathrm{DP}}\, \tau^p, \tag{129}$$

for some $p > 1$ and constant $C_{\mathrm{DP}} < \infty$.

In addition if the deterministic Hamiltonian bias bound (126) *holds for some $\gamma > 0$, then for all $k \geq 0$,*

$$\|\widetilde{V}_k - V^{(\alpha)}\|_\infty \leq e^{-\rho\tau k}\, \|\widetilde{V}_0 - V^{(\alpha)}\|_\infty + \frac{C_{\mathrm{DP}}\tau^p + \tau C_H\, \mathbb{E}[U^\gamma]}{1 - e^{-\rho\tau}}. \tag{130}$$

Proof. For any V,

$$\|\widetilde{T}_\tau^{(\alpha)}(V) - (V + \tau H^{(\alpha)}(V))\|_\infty = \tau\|\bar{H}_U^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \leq \tau C_H\, \mathbb{E}[U^\gamma]$$

by Lemma G.1. Combining with the local semigroup mismatch (129) gives

$$\|\widetilde{T}_\tau^{(\alpha)}(V) - \Phi_\tau^{(\alpha)}(V)\|_\infty \leq C_{\mathrm{DP}}\tau^p + \tau C_H\, \mathbb{E}[U^\gamma].$$

Again, using contraction of $\Phi_\tau^{(\alpha)}$ and the fixed-point identity $\Phi_\tau^{(\alpha)}(V^{(\alpha)}) = V^{(\alpha)}$ yields the recursion

$$\|\widetilde{V}_{k+1} - V^{(\alpha)}\|_\infty \leq e^{-\rho\tau}\|\widetilde{V}_k - V^{(\alpha)}\|_\infty + C_{\mathrm{DP}}\tau^p + \tau C_H\, \mathbb{E}[U^\gamma],$$

and a similar inductive argument proves (130). □

Corollary G.3. *Based on the error estimates from Section E, we can naturally extends to random time settings under the following 3 scenarios:*

***(1) Finite-horizon Picard-Hamiltonian iterations with q_V^u under standard assumption**. If $\|H_u^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \le Cu^{1/2}$ and $\|\Phi_\tau^{(\alpha)}(V) - (V + \tau H^{(\alpha)}(V))\|_\infty \le C\tau^{3/2}$, then* (130) *holds with $(p, \gamma) = (3/2, 1/2)$ and the random bias term becomes $\tau\mathbb{E}[U^{1/2}]$.*

***(2) Finite-horizon Picard-Hamiltonian iterations with q_V^u under smoother assumption**. If $\|H_u^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \le Cu$ and the semigroup mismatch is $\mathcal{O}(\tau^2)$, then $(p, \gamma) = (2, 1)$ and the random bias term becomes $\tau\mathbb{E}[U]$.*

***(3) Richardson Picard-Hamiltonian iterations**. If $\|\tilde{H}_u^{(\alpha)}(V) - H^{(\alpha)}(V)\|_\infty \le Cu^2$ and the semigroup mismatch is $\mathcal{O}(\tau^2)$, then $(p, \gamma) = (2, 2)$ and the random bias term becomes $\tau\mathbb{E}[U^2]$.*

G.1 q-CONVERGENCE

The Hamiltonian transfer in Lemma G.1 is immediate because the bias bound depends only on positive moments $\mathbb{E}[U^\gamma]$. For the advantage-rate q_V^u, an additional stability condition is needed because the finite-horizon definition contains a division by u, making the map $V \mapsto q_V^u$ Lipschitz with constant of order $1/u$.

Lemma G.4 (Random-time Lipschitz stability of q_V^U)**.** *Fix $\beta > 0$ and for $u \in (0, 1]$. Let U be a random variable taking values in $(0, 1]$, independent of the state and control. Then for any bounded V, W,*

$$\mathbb{E}\big[\|q_V^U - q_W^U\|_\infty\big] \le 2\,\mathbb{E}\Big[\frac{1}{U}\Big]\,\|V - W\|_\infty, \tag{131}$$

and

$$\mathbb{E}\big[\|\tilde{q}_V^U - \tilde{q}_W^U\|_\infty\big] \le 10\,\mathbb{E}\Big[\frac{1}{U}\Big]\,\|V - W\|_\infty. \tag{132}$$

In particular, if $U \ge u_{\min} > 0$ almost surely, then the above bounds hold with $\mathbb{E}[1/U] \le 1/u_{\min}$.

Proof. Fix $u \in (0, 1]$. Using definitions of q^u and cancelling $r(x, a)$, we have:

$$|q_V^u(x,a) - q_W^u(x,a)| \le \frac{1}{u}\Big(e^{-\beta u}\big|\mathbb{E}_x^a[V(X_u^a) - W(X_u^a)]\big| + |V(x) - W(x)|\Big) \le \frac{2}{u}\|V - W\|_\infty.$$

Taking the supremum over (x, a) yields $\|q_V^u - q_W^u\|_\infty \le \frac{2}{u}\|V - W\|_\infty$. Evaluating at $u = U$ and taking expectation gives (131). For Richardson, apply the same bound to u and $u/2$:

$$\|\tilde{q}_V^u - \tilde{q}_W^u\|_\infty \le 2\|q_V^{u/2} - q_W^{u/2}\|_\infty + \|q_V^u - q_W^u\|_\infty \le \Big(2 \cdot \frac{4}{u} + \frac{2}{u}\Big)\|V - W\|_\infty = \frac{10}{u}\|V - W\|_\infty,$$

and again evaluate at $u = U$ and take expectation to obtain (132). □

Theorem G.5 (Random holding times: averaged q-convergence)**.** *Let $V^* := V^{(\alpha)}$ be the optimal value function. Assume the random-time averaged Picard iteration $(\widetilde{V}_k)_{k\ge0}$ satisfy the value error bound*

$$\|\widetilde{V}_k - V^*\|_\infty \le e^{-\rho\tau k}\|\widetilde{V}_0 - V^*\|_\infty + \frac{C_{\mathrm{DP}}\tau^p + \tau C_H\,\mathbb{E}[U^\gamma]}{1 - e^{-\rho\tau}}. \tag{133}$$

If $\mathbb{E}[1/U] < \infty$, then the corresponding random-time rates obey

$$\mathbb{E}\big[\|q_{\widetilde{V}_k}^U - q(V^*)\|_\infty\big] \le 2\,\mathbb{E}\Big[\frac{1}{U}\Big]\,\|\widetilde{V}_k - V^*\|_\infty + C_q\,\mathbb{E}[U^\gamma], \tag{134}$$

and, if Richardson is used,

$$\mathbb{E}\big[\|\tilde{q}_{\widetilde{V}_k}^U - q(V^*)\|_\infty\big] \le 10\,\mathbb{E}\Big[\frac{1}{U}\Big]\,\|\widetilde{V}_k - V^*\|_\infty + C_q'\,\mathbb{E}[U^2]. \tag{135}$$

Proof. Decompose and apply triangle inequality:

$$\mathbb{E}\big[\|q_{\widetilde{V}_k}^U - q(V^*)\|_\infty\big] \le \mathbb{E}\big[\|q_{\widetilde{V}_k}^U - q_{V^*}^U\|_\infty\big] + \mathbb{E}\big[\|q_{V^*}^U - q(V^*)\|_\infty\big].$$

The first term is controlled by Lemma G.4. The second term is the modelling bias, which follows from the deterministic bias bound $\|q_{V^*}^u - q(V^*)\|_\infty \le C_q u^\gamma$ and averaging over U: $\mathbb{E}[\|q_{V^*}^U - q(V^*)\|_\infty] \le C_q\mathbb{E}[U^\gamma]$. The Richardson case is identical, using the Richardson modelling bias $\|\tilde{q}_{V^*}^u - q(V^*)\|_\infty \le C_q' u^2$ and the Lipschitz bound (132). □

Note. If $U \in [u_{\min}, 1]$ almost surely, then $\mathbb{E}[1/U] \le 1/u_{\min}$ and the q-bounds (134)–(135) hold with explicit constants. This matches the typical irregular-timestep setting where the environment enforces a minimum timestep.

H Regret bound and sample complexity

This section provides a refined statistical analysis of the single-critic framework. The argument separates the population approximation error from the statistical error incurred when fitting each critic update. This decomposition applies to both the first-order finite-horizon update and its higher-order variants. For the explicit rate calculation below, we use the sharper population approximation bound established in Theorem 4.2. Analogous bounds follow by substituting the corresponding population estimate for the particular update under consideration. The goal is to control the cumulative gap $\sum_{k=0}^{L-1} \|\hat{Q}_k - Q^{(\alpha)}\|_\infty$. The main technique that allows a more refined error analysis is a geometric argument that upgrades $L^2(\nu)$ control to uniform $\|\cdot\|_\infty$ control for Lipschitz functions on a compact domain.

Let $\mathcal{F}_{\tau,u}$ denote the corresponding exact population single-critic update, so that

$$Q_{k+1} = \mathcal{F}_{\tau,u}(Q_k),$$

and let $\hat{Q}_{k+1}$ denote its sample-based approximation.

Recall the statistical error $\mathrm{stat}_k := \|\hat{Q}_{k+1} - \mathcal{F}_{\tau,u}(\hat{Q}_k)\|_{L^2(\nu)}$. Also recall that thanks to Lemmas F.4 to F.6, we have:

$$\mathrm{stat}_k \;\lesssim\; \frac{1}{\sqrt{n_k}}\left(1 + \sqrt{\log d} + \sqrt{\log(1/\delta)}\right) \qquad \text{with probability} \ge 1-\delta, \tag{136}$$

Setup and assumptions Let $\mathcal{X} \subset \mathbb{R}^d$ ($d = d_{\mathcal{X}}$) be the state space and $\mathcal{A} \subset \mathbb{R}^{d_{\mathcal{A}}}$ the action space. Define the state–action domain with dimension m:

$$\mathcal{Z} := \mathcal{X} \times \mathcal{A}, \qquad m := d_{\mathcal{X}} + d_{\mathcal{A}}.$$

We write $z = (x, a) \in \mathcal{Z}$, equip $\mathcal{Z}$ with the Euclidean metric $d(z, z') := \|z - z'\|_2$, and denote the sup norm on $\mathcal{Z}$ by $\|f\|_\infty = \sup_{z\in\mathcal{Z}} |f(z)|$. Throughout, we fix an algorithmic step size $\tau > 0$ and a holding time $u > 0$, with $\gamma := e^{-\beta u}$.

Assumption H.1 (Compact domain and boundedness)**.** We have the following assumptions. First, we assume state-action space $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$ is compact and has diameter $\mathrm{diam}(\mathcal{Z}) \le R$ for some $R > 0$. Additionally, assume the rewards are bounded: $|r(x, a)| \le r_{\max}$

Assumption H.2 (Coverage density lower bound)**.** At each critic regression step k, training samples $Z_i = (X_i, A_i)$ are drawn i.i.d. from a distribution ν_k supported on $\mathcal{Z}$ whose density is bounded below by $\nu_{\min} > 0$ with respect to Lebesgue measure on $\mathbb{R}^m$. Equivalently, for any measurable $E \subseteq \mathcal{Z}$,

$$\nu_k(E) \;\ge\; \nu_{\min}\,\mathrm{Vol}(E).$$

Assumption H.3 (Lipschitz critic)**.** All critics considered are uniformly bounded and Lipschitz on $\mathcal{Z}$: there exist constants $B_Q, L_Q > 0$ such that for every iterate Q (population or learned),

$$\|Q\|_\infty \le B_Q, \qquad |Q(z) - Q(z')| \le L_Q\,\|z - z'\|_2, \qquad \forall z, z' \in \mathcal{Z}.$$

Moreover, for each fixed Q, the population mapping $\mathcal{F}_{\tau,u}(Q)$ is also Lipschitz with

$$\|\mathcal{F}_{\tau,u}(Q)\|_\infty \le B_F, \qquad \mathrm{Lip}\big(\mathcal{F}_{\tau,u}(Q)\big) \le L_F,$$

for constants B_F, L_F that do not depend on k. (The same holds for the Richardson mapping $\mathcal{F}^{\mathrm{Rich}}_{\tau,u}$.)

This assumption H.3 holds true by a similar argument to the proof of Lemma F.9

H.1 A norm-transfer argument from L^2 to $\|\cdot\|_\infty$

We start with a few lemmas from a standard norm-transfer argument.

Lemma H.4 (Uniform lower bound on ball mass)**.** *Under assumption H.2, for any $z \in \mathcal{Z}$ and any $r > 0$ with $B_r(z) \subseteq \mathcal{Z}$,*

$$\nu\big(B_r(z)\big) \;\geq\; \nu_{\min}\,\mathrm{Vol}\big(B_r(z)\big) \;=\; \nu_{\min}\,c_m\,r^m, \tag{137}$$

where $c_m := \mathrm{Vol}(B_1(0))$ is the volume of the unit ball in $\mathbb{R}^m$.

Lemma H.5 ($L^2(\nu)$ to sup-norm)**.** *Let $g : \mathcal{Z} \to \mathbb{R}$ be L_g-Lipschitz and suppose $\|g\|_{L^2(\nu)} \leq \varepsilon$. Under Assumption H.2, one has*

$$\|g\|_\infty \;\leq\; \Big(\frac{2^{m+2}L_g^m}{\nu_{\min}c_m}\Big)^{\frac{1}{m+2}}\,\varepsilon^{\frac{2}{m+2}}. \tag{138}$$

Proof. Let $z^* \in \arg\max_{z\in\mathcal{Z}}|g(z)|$ and set $M := \|g\|_\infty = |g(z^*)|$. Choose $r := M/(2L_g)$. By Lipschitzness, for any $z \in B_r(z^*)$,

$$|g(z)| \geq |g(z^*)| - L_g\|z - z^*\|_2 \geq M - L_g r = M/2.$$

Therefore,

$$\varepsilon^2 \;\geq\; \int_{\mathcal{Z}} g(z)^2\,\nu(dz) \;\geq\; \int_{B_r(z^*)} g(z)^2\,\nu(dz) \;\geq\; (M/2)^2\,\nu\big(B_r(z^*)\big).$$

By Lemma H.4, $\nu(B_r(z^*)) \geq \nu_{\min}c_m r^m = \nu_{\min}c_m(M/(2L_g))^m$. Substituting gives

$$\varepsilon^2 \;\geq\; \frac{M^2}{4}\,\nu_{\min}c_m\Big(\frac{M}{2L_g}\Big)^m \;=\; \frac{\nu_{\min}c_m}{2^{m+2}L_g^m}\,M^{m+2}.$$

Rearranging yields Equation (138). □

Aggregate bounds. Now we summarize the inequality ingredients for the main bound. Because of Lemma H.5, there exists a universal constant C_∞ so that, with probability of at least $1-\delta$:

$$\|\hat{Q}_{k+1} - \mathcal{F}_{\tau,u}(\hat{Q}_k)\|_\infty \leq C_\infty\,\mathrm{stat}_k^{\frac{2}{m+2}} \leq C_\infty\left(\frac{1+\sqrt{\log d}+\sqrt{\log(1/\delta)}}{\sqrt{n_k}}\right)^{\frac{2}{m+2}} \tag{139}$$

Additionally, recall that $Q^{(\alpha)}$ is the optimal soft critic $Q^{(\alpha)}(x,a) = V^{(\alpha)}(x) + q_{V^{(\alpha)}}(x,a)$. Under the refined population estimate of Theorem 4.2, there exist $\rho > 0$ and $C > 0$ such that the approximation component admits the bound:

$$\|Q_k - Q^{(\alpha)}\|_\infty \leq \frac{e^{-\rho\tau k}}{u}\,\|Q_0 - Q^{(\alpha)}\|_\infty + C\big(\tau/u + u\big), \tag{140}$$

H.2 Main bound

Define the one-step *sup-norm statistical deviation*

$$\zeta_k := \big\|\hat{Q}_{k+1} - \mathcal{F}_{\tau,u}(\hat{Q}_k)\big\|_\infty, \tag{141}$$

Lemma H.6 (Cumulative sup-norm bound)**.** *Suppose the update operator $\mathcal{F}_{\tau,u}$ is $(1-\alpha)$-Lipschitz for $\alpha < 1$ where α depends on τ and u. Under Assumption H.1 and H.2 (assumption H.3 is satisfied), for any $L \geq 1$,*

$$\sum_{k=0}^{L-1}\|\hat{Q}_k - Q^{(\alpha)}\|_\infty \leq \frac{1}{(1-e^{-\rho\tau})u}\|\hat{Q}_0 - Q^{(\alpha)}\|_\infty + CL\big(\tau/u+u\big) + \frac{1}{\alpha}\left(\sum_{k=0}^{L-1}\zeta_k\right) \tag{142}$$

Proof. The bias (algorithmic) error $\big\|Q_k - Q^{(\alpha)}\big\|_\infty$ makes up the first 2 terms in Equation (142). This is clear from Equation (140). Now we bound the statistical error with $E_k := \Big\|\hat{Q}_k - Q_k\Big\|_\infty$. We have:

$$\begin{aligned} E_{k+1} = \Big\|\hat{Q}_{k+1} - \mathcal{F}_{\tau,u}(\hat{Q}_k)\Big\|_\infty &\leq \Big\|\hat{Q}_{k+1} - \mathcal{F}_{\tau,u}(\hat{Q}_k)\Big\|_\infty + \Big\|\mathcal{F}_{\tau,u}(\hat{Q}_k) - \mathcal{F}_{\tau,u}(Q_k)\Big\|_\infty \\ &\leq \zeta_k + (1-\alpha)\Big\|\hat{Q}_k - Q_k\Big\| = \zeta_k + (1-\alpha)E_k \end{aligned}$$

Hence, through an inductive argument, the running gap $\sum_{k=0}^{L-1}E_k$ is bounded by $\sum_k \zeta_k(1 + (1-\alpha) + \cdots + (1-\alpha)^{L-1})$, leading to the third term of Equation (142) □

Note that the terms in the update operators (see Equation (117)) can be rearranged so that it consists of 3 components with Lipschitz coefficients $\big(1-\tau\big)$, $\dfrac{\gamma\tau}{u}$ and $\big(\tau-\dfrac{\tau}{u}\big)$. In case if $u \geq 1$, this yields a $(1-\alpha)$ Lipchitz operator with $\alpha = (1-\gamma)\dfrac{\tau}{u}$

Theorem H.7 (Sample complexity and regret bounds)**.** *Given the probability confidence level $\delta \in (0,1)$ and maximum number of iteration L_{max}. Suppose the update operator is $(1-\alpha)$-Lipschitz with $\alpha \geq C_0\dfrac{\tau}{u}$. Then there exists (q-function discretization) timestep u, the Hamiltonian flow discretization step τ, a sequence of numbers of samples $\{n_k\}$, and an universal constants C_1 and C_2 (independent of L, L_{max}, τ or u), such that for every iteration $L_{max}/2 \leq L \leq L_{max}$, with probability at least $1-\delta$, the running gap satisfies:*

$$\sum_{k=0}^{L-1} \|\hat{Q}_k - Q^{(\alpha)}\|_\infty \leq C_1 L^{3/4} \tag{143}$$

Additionally, in both cases, the total sample size in the first L iteration satisfies

$$K = K(L) := \sum_{k=0}^{L-1} n_k = C_2 L^{m+3}. \tag{144}$$

Consequently, for any given fixed L, ignore the log-terms, we get $\mathrm{Regret}(K) = \mathcal{O}(K^{\frac{4m+11}{4m+12}})$ ***(sub-linear regret** rate).*

Proof. Choose $\tau = L_{\max}^{-1/2}$, $u = L_{\max}^{-1/4}$. Choose the number of samples $n_k = (k+1)^{m+2}$ for all $k \geq 0$.

Now we choose per-iteration confidence $\delta_k := \delta/L$ for $k \in \overline{0, L-1}$. Through the statistical bound derived above, with probability $\geq 1-\delta$ (via union bound over k using $\delta_k = \delta/L$), we have for all $k \in \overline{0, L-1}$:

$$\zeta_k \leq \Big(\mathrm{stat}_k\Big)^{\frac{2}{m+2}} \leq C_\infty \left(\frac{1+\sqrt{\log d}+\sqrt{\log(L/\delta)}}{\sqrt{n_k}}\right)^{\frac{2}{m+2}}.$$

Now we analyze each error term within the large bracket of Equation (142). First, with $n_k = (k+1)^{m+2}$, the right-hand side for the inequality of ζ_k can be upper-bounded by $C_3\big(\log(L/\delta)^{\frac{1}{m+2}}/(k+1)\big)$. Hence $\sum_{k=0}^{L-1}\zeta_k \leq C_4\big(\log(L/\delta)^{\frac{1}{m+2}} \times \log L\big)$, and the statistical part is then only multiply by $1/\alpha = u/\tau = C_5 L^{1/4}$.

Second, due to the choice of τ and u, the bias part Equation (142) from can be bounded by $C_6\dfrac{1}{\tau u} + CL(\tau/u + u) = C_7 L^{3/4}$, which dominates the statistical part and yields the overall error rate in Equation (143). The sample complexity follows from $\sum_{k=1}^{L} k^{m+2} \leq C_1 L^{m+3}$ for some universal constant C_1.

Finally, regarding the regret bound, which can be defined by sum of gaps $Q^{(\alpha)}(Z) - \hat{Q}_k(Z)$ over starting points Z of all episodes. Thus, up to iteration L (for any $L \leq L_{\max}$), the regret can be over-estimated by:

$$\begin{aligned}\mathrm{Regret} &\leq \sum_{\text{all samples } Z} \mathbb{E}\left\|\hat{Q}_k(Z) - Q^{(\alpha)}(Z)\right\| \leq \sum_{k=0}^{L-1} n_k \left\|\hat{Q}_k - Q^{(\alpha)}\right\|_\infty \\ &\leq \left(\max_{k\in\overline{0,L-1}} n_k\right) \sum_{k=0}^{L-1} \|\hat{Q}_k - Q^{(\alpha)}\|_\infty \leq C_1 L^{m+2} L^{3/4} = \mathcal{O}(L^{m+11/4})\end{aligned}$$

As $K = \mathcal{O}(L^{m+3})$, $\mathcal{O}(L^{m+11/4}) = \mathcal{O}(K^{\frac{m+11/4}{m+3}}) = \mathcal{O}(K^{\frac{4m+11}{4m+12}})$, yielding the desired regret bound. Here all C_i's are universal constants independent of L, $L_{\max}$, τ or u.

The same decomposition applies to the first-order update Equation (28) and other finite-horizon variants after replacing the population approximation term above by the corresponding bound for that update. □

Notes. For random time step $U \in (0, 1]$, independent of (x, a), the only change is the bias term, which becomes a moment of U: $\mathbb{E}[\mathrm{bias}(U)]$. In particular: $\mathrm{bias}(U) = \mathbb{E}[U^{1/2}]$ for standard case, and $\mathrm{bias}(U) = \mathbb{E}[U^2]$ for Richardson-version. Additionally, an improved variant with sharper regret and complexity guarantees will be studied in subsequent work. The proof techniques developed here extend naturally to adaptive step sizes $(\tau_k)_{k\geq 0}$; we also plan to derive explicit rates for such schedules in the future.

I Further Theoretical Details For Main Paper

I.1 Further explanations for the main paper

Variational representation for Hamiltonian. For the Hamiltonian function, by a standard variational argument (see Lemma C.6), one can equivalently write:

$$H^{(\alpha)}(V)(x) = \sup_{\pi(\cdot|x)} \int_{\mathcal{A}} \Big(r(x,a) + (L^a V)(x) - \beta V(x) - \alpha \log \pi(a \mid x) \Big) \pi(da \mid x) \tag{145}$$

As $\alpha \to 0$, $H^{(\alpha)}$ recovers $H^{(0)}$ and, for $\alpha > 0$, the policy π that maximize the above variational form is exactly the Boltzmann policy

$$\pi_V^{(\alpha)}(a \mid x) \propto \exp\Big(\tfrac{1}{\alpha} \big(r(x,a) + (L^a V)(x) - \beta V(x) \big) \Big) \tag{146}$$

Distinction between u and τ. We emphasize that our framework separates two distinct discretizations: (i) the environment produces transitions with potentially irregular durations u, and (ii) the algorithm updates V using a flow step size τ, which controls value-iteration stability and convergence and is not tied to u. This separation is one reason the method remains stable under small or irregular time steps. Additionally, in this work, we make two concrete modeling choices for simplicity and scalability. First, we discretize the value flow Equation (21) using a single forward-Euler (Picard) step. More accurate ODE solvers (e.g., higher-order Runge–Kutta schemes) could be substituted to improve numerical stability. Second, we implement the decoupled iteration with a single critic by using the algebraic representation $Q = V + 1 \cdot q$ in Equation (27). This choice is viewed as selecting a nominal unit time scale for combining a value-like term and an instantaneous rate term. More generally, one may consider $Q = V + c\, q$ for other scalings $c > 0$, which may trade off learning signal, conditioning, and empirical performance. Exploring such optimal discretization and scaling strategies is the scope for future work.

I.2 Discrete MDP with noise converge to SDE

The usual Markov decision process (MDP) discretized with a fixed time step $\Delta > 0$ has the form:

$$X^{\Delta}_{t+\Delta} = X^{\Delta}_t + f^{\Delta}(X^{\Delta}_t, a_t), \qquad a_t \sim \pi_\alpha(\cdot \mid X^{\Delta}_t) \tag{147}$$

where $t \in \{0, \Delta, 2\Delta, \dots\}$, $X^{\Delta}_t \in \mathbb{R}^d$, and π_α denotes an exploration policy. Through classical limit arguments, Munos [15] shows that if the increment is approximately linear in Δ, i.e, $f^{\Delta}(x, a) = f(x, a)\,\Delta + o(\Delta)$, then, as $\Delta \to 0$, the discrete process converges to a deterministic trajectory x_t governed by the ODE:

$$\dot{x}_t = f_\alpha(x_t), \qquad f_\alpha(x) := \mathbb{E}_{a\sim\pi_\alpha(\cdot|x)}\big[f(x,a)\big] \tag{148}$$

where action randomization averages into the drift. To model stochasticity beyond randomized actions, we augment the transition with a mean-zero perturbation,

$$X^{\Delta}_{t+\Delta} = X^{\Delta}_t + f^{\Delta}(X^{\Delta}_t, a_t) + \xi^{\Delta}_t \tag{149}$$

Under diffusion scaling (e.g., $\mathrm{Var}(\xi^{\Delta}_t) = \mathcal{O}(\Delta)$), the accumulated fluctuations converge to Brownian motion (Donsker's invariance principle; 5), and the continuous-time limit becomes a controlled diffusion. This motivates the Itô SDE model used throughout the paper. Below we rigorously prove this convergence.

Lemma I.1 (Convergence of MDP to SDE). *Fix $T > 0$ and let the time step $\Delta_n \to 0$. Define grid times $t_k = k\Delta_n$ and $\bar{t} := t_k$ be the left endpoint mapping for $t \in [t_k, t_{k+1})$. Assume $f(\cdot, u)$ and $\sigma(\cdot, u)$ are globally Lipschitz and of linear growth in x, and let X_t solve the SDE:*

$$dX_t = f(X_t, U_t)\,dt + \sigma(X_t, U_t)\,dW_t, \qquad X_0 = x.$$

Define (discretized) MDP process $X_k^{(n)}$ with discretized time step Δ_n:

$$X_{k+1}^{(n)} = X_k^{(n)} + f(X_k^{(n)}, U_{t_k})\Delta_n + \sigma(X_k^{(n)}, U_{t_k})\big(W_{t_{k+1}} - W_{t_k}\big), \qquad X_0^{(n)} = x,$$

and let $X_t^{(n)} := X_k^{(n)}$ for $t \in [t_k, t_{k+1})$ (piecewise-constant interpolation), and similarly $U_{\bar{t}} := U_{t_k}$. Then there exists a constant $C = C(T, L, K, x)$ such that

$$\mathbb{E}\Big[\sup_{0\le t\le T} \|X_t^{(n)} - X_t\|^2\Big] \le C\,\Delta_n.$$

In particular, $\sup_{t\le T} \|X_t^{(n)} - X_t\| \to 0$ in L^2 (and hence in probability).

Proof. First by the proof of Lemma C.5, $\mathbb{E}\|X_t - X_s\|^2 \le C|t-s|$. Therefore, for any $u \in [0,T]$ with $\bar{u}$ its left gridpoint, $|u - \bar{u}| \le \Delta_n$ and $\mathbb{E}\|X_u - X_{\bar{u}}\|^2 \le C\,\Delta_n$. Now, the piecewise-constant Euler process can be written as

$$X_t^{(n)} = x + \int_0^t f(X_{\bar{s}}^{(n)}, U_{\bar{s}})\,ds + \int_0^t \sigma(X_{\bar{s}}^{(n)}, U_{\bar{s}})\,dW_s.$$

Subtracting the SDE gives the error process $e_t := X_t^{(n)} - X_t$:

$$e_t = \int_0^t \big[f(X_{\bar{s}}^{(n)}, U_{\bar{s}}) - f(X_s, U_s)\big]\,ds + \int_0^t \big[\sigma(X_{\bar{s}}^{(n)}, U_{\bar{s}}) - \sigma(X_s, U_s)\big]\,dW_s.$$

Let $E_t := \sup_{0\le r\le t} \|e_r\|$.

Drift term. With $g_s = f(X_{\bar{s}}^{(n)}, U_{\bar{s}}) - f(X_s, U_s)$, using Lipschitzness, we get:

$$\|g_s\| \le L\|X_{\bar{s}}^{(n)} - X_s\| \le L\big(\|X_{\bar{s}}^{(n)} - X_{\bar{s}}\| + \|X_{\bar{s}} - X_s\|\big) \le L\big(E_s + \|X_{\bar{s}} - X_s\|\big).$$

As a result,

$$\mathbb{E}\Big[\sup_{0\le r\le t} \Big\| \int_0^r g_s\,ds\Big\|^2\Big] \le C\int_0^t \mathbb{E}[E_s^2]\,ds + C\int_0^t \mathbb{E}\|X_s - X_{\bar{s}}\|^2\,ds. \tag{150}$$

Diffusion term. By the Burkholder–Davis–Gundy inequality (for $p = 2$),

$$\mathbb{E}\Big[\sup_{0\le r\le t} \Big\| \int_0^r h_s\,dW_s\Big\|^2\Big] \le C_{\mathrm{BDG}}\,\mathbb{E}\int_0^t \|h_s\|_F^2\,ds.$$

With $h_s = \sigma(X_{\bar{s}}^{(n)}, U_{\bar{s}}) - \sigma(X_s, U_s)$ and Lipschitzness,

$$\|h_s\|_F \le L\|X_{\bar{s}}^{(n)} - X_s\| \le L\big(E_s + \|X_{\bar{s}} - X_s\|\big),$$

so that

$$\mathbb{E}\Big[\sup_{0\le r\le t} \Big\| \int_0^r h_s\,dW_s\Big\|^2\Big] \le C\int_0^t \mathbb{E}[E_s^2]\,ds + C\int_0^t \mathbb{E}\|X_s - X_{\bar{s}}\|^2\,ds. \tag{151}$$

Now combining (150)–(151) together with triangle inequality,

$$\mathbb{E}[E_t^2] \le C\int_0^t \mathbb{E}[E_s^2]\,ds + C\int_0^t \mathbb{E}\|X_s - X_{\bar{s}}\|^2\,ds.$$

As proved earlier, $\mathbb{E}\|X_s - X_{\bar{s}}\|^2 \le C\Delta_n$ for all $s \le T$, so that

$$\mathbb{E}[E_t^2] \le C\int_0^t \mathbb{E}[E_s^2]\,ds + C\,t\,\Delta_n.$$

Finally, Gronwall's lemma gives $\mathbb{E}[E_T^2] \le C\,\Delta_n$, proving the claim with the error go to 0 as $\Delta_n \to 0$. □

Note that instead of $\big(W_{t_{k+1}} - W_{t_k}\big)$, we can replace it by i.i.d normal random variable ξ_{t_k}, and all the inequalities derived in the proof still holds in expectation as they has the same distribution due to the property of the Brownian motion.

I.3 Exact Richardson single-critic update equation.

Lemma I.2 (Algebraic derivation of the Richardson single-critic update)**.** *Fix $u > 0$ and define $\gamma := e^{-\beta u}$ and $\gamma_{1/2} := e^{-\beta u/2}$. Let V_k be updated by the entropy-regularized expectation step*

$$V_{k+1}(x) = (1-\tau)V_k(x) + \tau S_k(x), \qquad S_k(x) := \mathbb{E}_{a\sim\pi_k(\cdot|x)}\big[\tilde{Q}_k^{\mathrm{R}}(x,a)\big], \tag{152}$$

where $Q_k^{\mathrm{R}}(x,a) := V_k(x) + \tilde{q}^u_{V_k}(x,a)$ and $\tilde{Q}_k^{\mathrm{R}}(x,a) := Q_k^{\mathrm{R}}(x,a) - \alpha \log \pi_k(a \mid x)$. Then the induced single-critic iteration satisfies the closed-form update:

$$\begin{aligned} Q_{k+1}^{\mathrm{R}}(x,a) &= (1-\tau)Q_k^{\mathrm{R}}(x,a) + \tau r(x,a) + \tau S_k(x) \\ &\quad + \frac{\tau}{u}\Big(4\gamma_{1/2}\,\mathbb{E}_x^a[S_k(X_{u/2})] - \gamma\,\mathbb{E}_x^a[S_k(X_u)] - 3S_k(x)\Big). \end{aligned} \tag{153}$$

Proof. We start with definition of $q_V^{u/2}$:

$$q_V^{u/2}(x,a) = \frac{\gamma_{1/2}\mathbb{E}_x^a[V(X_{u/2})] - V(x)}{u/2} + r(x,a) = \frac{2(\gamma_{1/2}\mathbb{E}_x^a[V(X_{u/2})] - V(x))}{u} + r(x,a).$$

Hence

$$\begin{aligned} \tilde{q}_V^u(x,a) &= 2q_V^{u/2}(x,a) - q_V^u(x,a) \\ &= \frac{4\gamma_{1/2}\mathbb{E}_x^a[V(X_{u/2})] - \gamma\mathbb{E}_x^a[V(X_u)] - 3V(x)}{u} + r(x,a). \end{aligned} \tag{154}$$

Now apply Equation (154) to V_{k+1}:

$$\tilde{q}^u_{V_{k+1}}(x,a) = \frac{4\gamma_{1/2}\mathbb{E}_x^a[V_{k+1}(X_{u/2})] - \gamma\mathbb{E}_x^a[V_{k+1}(X_u)] - 3V_{k+1}(x)}{u} + r(x,a). \tag{155}$$

Substitute the value update Equation (152) inside each term. Since the update is pointwise in the state, for any random state Y we have $V_{k+1}(Y) = (1-\tau)V_k(Y) + \tau S_k(Y)$, and therefore

$$\mathbb{E}_x^a[V_{k+1}(X_h)] = (1-\tau)\mathbb{E}_x^a[V_k(X_h)] + \tau\mathbb{E}_x^a[S_k(X_h)], \qquad h \in \{u/2, u\}.$$

Also $V_{k+1}(x) = (1-\tau)V_k(x) + \tau S_k(x)$. Plugging these into Equation (155) and collecting the corresponding terms gives:

$$\begin{aligned} \tilde{q}^u_{V_{k+1}}(x,a) &= (1-\tau)\frac{4\gamma_{1/2}\mathbb{E}_x^a[V_k(X_{u/2})] - \gamma\mathbb{E}_x^a[V_k(X_u)] - 3V_k(x)}{u} \\ &\quad + \tau\frac{4\gamma_{1/2}\mathbb{E}_x^a[S_k(X_{u/2})] - \gamma\mathbb{E}_x^a[S_k(X_u)] - 3S_k(x)}{u} + r(x,a). \end{aligned} \tag{156}$$

The first fraction is exactly $\tilde{q}^u_{V_k}(x,a) - r(x,a)$ by Equation (154). Thus

$$\begin{aligned} \tilde{q}^u_{V_{k+1}}(x,a) &= (1-\tau)\big(\tilde{q}^u_{V_k}(x,a) - r(x,a)\big) + r(x,a) \\ &\quad + \frac{\tau}{u}\Big(4\gamma_{1/2}\mathbb{E}_x^a[S_k(X_{u/2})] - \gamma\mathbb{E}_x^a[S_k(X_u)] - 3S_k(x)\Big) \\ &= (1-\tau)\tilde{q}^u_{V_k}(x,a) + \tau r(x,a) + \frac{\tau}{u}\Big(4\gamma_{1/2}\mathbb{E}_x^a[S_k(X_{u/2})] - \gamma\mathbb{E}_x^a[S_k(X_u)] - 3S_k(x)\Big). \end{aligned} \tag{157}$$

Finally, add the value update $V_{k+1}(x) = (1-\tau)V_k(x) + \tau S_k(x)$ to both sides, we get

$$\begin{aligned} Q_{k+1}^{\mathrm{R}}(x,a) &= (1-\tau)V_k(x) + \tau S_k(x) + (1-\tau)\tilde{q}^u_{V_k}(x,a) + \tau r(x,a) + \frac{\tau}{u}\Big(\cdots\Big) \\ &= (1-\tau)\big(V_k(x) + \tilde{q}^u_{V_k}(x,a)\big) + \tau r(x,a) + \tau S_k(x) + \frac{\tau}{u}\Big(\cdots\Big) \\ &= (1-\tau)Q_k^{\mathrm{R}}(x,a) + \tau r(x,a) + \tau S_k(x) + \frac{\tau}{u}\Big(4\gamma_{1/2}\mathbb{E}_x^a[S_k(X_{u/2})] - \gamma\mathbb{E}_x^a[S_k(X_u)] - 3S_k(x)\Big), \end{aligned}$$

which is exactly Equation (153). □

I.4 Decoupled continuous-time policy evaluation

For a fixed policy π, standard stochastic control arguments imply that the value function $V^{(\alpha)}(\cdot;\pi)$ in (45) satisfies the linear elliptic PDE:

$$\beta V^{(\alpha)}(x;\pi) - (L^{\pi}V^{(\alpha)}(\cdot;\pi))(x) - \tilde{r}^{(\alpha)}(x,\pi) = 0 \tag{158}$$

Fix a Markov policy $\pi(a \mid x)$ and temperature $\alpha \geq 0$. For a value function $V \in \mathcal{D} \subset C_b^2(\mathbb{R}^d)$, define the *instantaneous q-rate* for (π, α) as

$$q_V^{(\pi,\alpha)}(x,a) := r(x,a) - \alpha \log \pi(a \mid x) + (L^a V)(x) - \beta V(x). \tag{159}$$

This corresponds to the infinitesimal temporal-difference

$$q_V^{(\pi,\alpha)}(x,a) = \lim_{\Delta t \to 0} \frac{e^{-\beta \Delta t}\mathbb{E}[V(X_{\Delta t})] - V(x)}{\Delta t} + r(x,a) - \alpha \log \pi(a \mid x) \tag{160}$$

This policy-dependent q-function $q_V^{(\pi,\alpha)}$ can be estimated from short rollouts by:

$$q_V^{u,(\pi,\alpha)}(x,a) := \frac{e^{-\beta u}V(X_{t+u}) - V(x)}{u} + r(x,a) - \alpha \log \pi(a \mid x) \tag{161}$$

Given $q_V^{(\pi,\alpha)}$, the policy-evaluation PDE satisfies

$$(L^{\pi}V)(x) + \tilde{r}^{(\alpha)}(x,\pi) - \beta V(x) = 0$$

And can be equivalently written as:

$$\int_{\mathcal{A}} q_V^{(\pi,\alpha)}(x,a)\, \pi(da \mid x) \;=\; 0, \qquad \forall x. \tag{162}$$

Following a similar idea as in Section 3, we obtain the following flow update for V under a fixed π:

$$\frac{d}{d\tau} V_\tau(x) \;=\; \int_{\mathcal{A}} q_{V_\tau}^{(\pi,\alpha)}(x,a)\, \pi(da \mid x). \tag{163}$$

A single explicit Euler step of (163) with step size $\tau > 0$ is

$$T_\tau^{(\pi,\alpha)}(V)(x) := V(x) + \tau \int_{\mathcal{A}} q_V^{(\pi,\alpha)}(x,a)\, \pi(da \mid x), \tag{164}$$

This yields an analogous Picard–Hamiltonian iteration for policy evaluation. The corresponding theoretical guarantees follow by applying the same argument as in the original setting, with only notational changes.

J Further experiment details

J.1 Visualization on the trading task

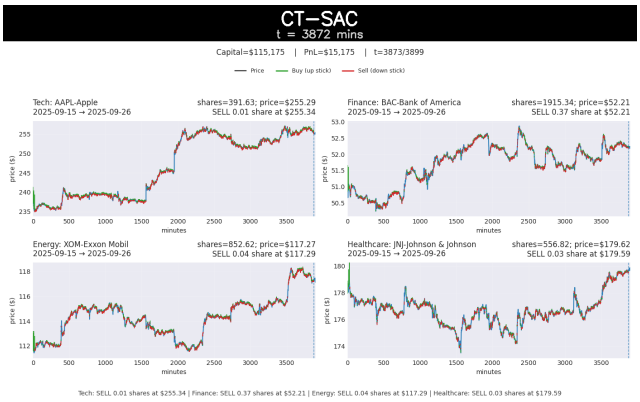
We provide additional (qualitative) visualizations for the trading task in Figure 7. These plots complement the quantitative metrics by illustrating representative trajectories under the same evaluation market paths.

J.2 Visualization on control tasks

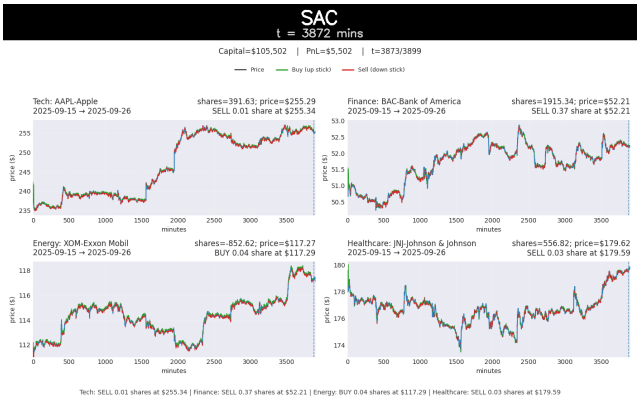
We provide additional qualitative results on the control tasks under two evaluation settings.

Irregular-time evaluation. We first evaluate on a held-out irregular-time setting and compare CT-SAC against (i) continuous-time baselines and (ii) discrete-time baselines. See Figures 8 and 9 for the corresponding images.

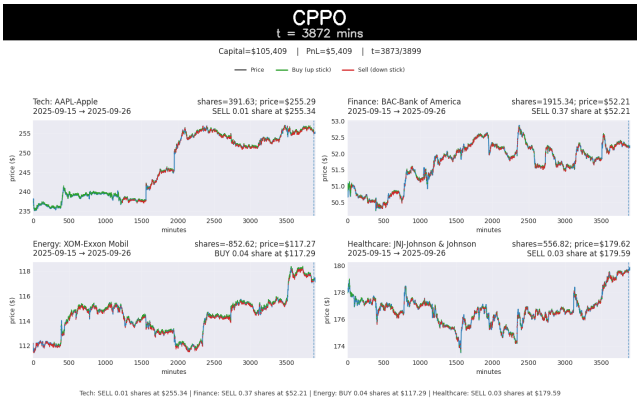
Regular-time evaluation. We also evaluate the same policies under the standard regular-time setting (which differs from the irregular-time training distribution) and again compare against continuous-time and discrete-time baselines. See Figures 10 and 11 for qualitative results.



(a) CT-SAC on trading task



(b) SAC on trading task

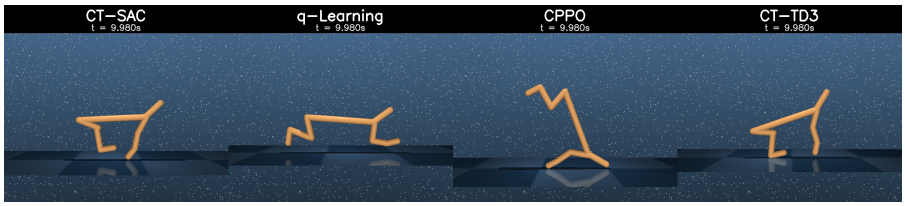


(c) CPPO on trading task

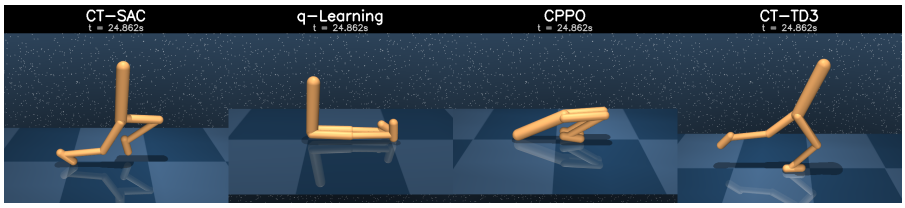
Figure 7: CT-SAC vs. two competitive baselines SAC and CPPO on trading task with two-week trading episode under irregular time steps. During these 2-weeks, CT-SAC made a profit of $15,000 while other two methods only make about $5,500.

J.3 Compute Resources

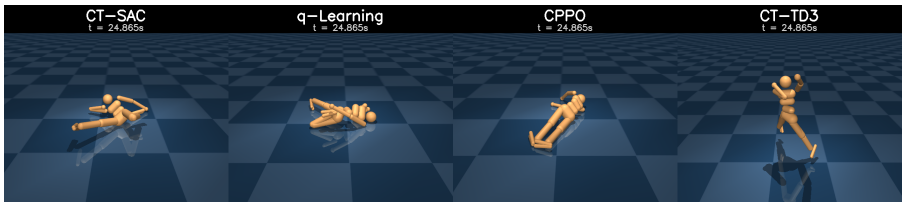
All experiments were executed on High-Performance Computing infrastructure. We used a *mixture* of CPU-only and GPU-accelerated nodes with different hardware characteristics in order to support both large-scale sweeps and computationally intensive deep RL training. Each run used a single node. We report details in Table 6.



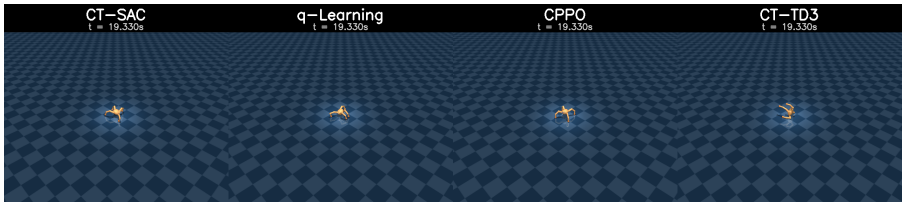
(a) Cheetah



(b) Walker



(c) Humanoid



(d) Quadruped

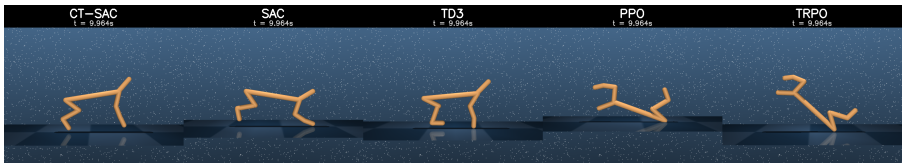
Figure 8: CT-SAC vs. continuous-time baselines under irregular time steps.

Table 6: Compute resources used in our experiments. "Cores" and "Memory" are per node.

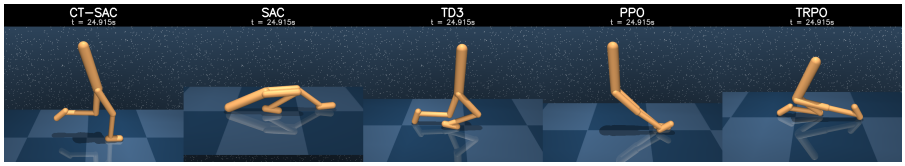
Node class	**CPU vendor (cores)**	**GPU vendor (count × memory)**	**Host memory**
CPU-only (high core)	NVIDIA (144)	–	237 GB
GPU node (single GPU)	NVIDIA (72)	NVIDIA (1×96 GB)	116 GB
CPU-only (general)	AMD (128)	–	256 GB
GPU node (multi-GPU)	AMD (128)	NVIDIA (3×40 GB)	256 GB
CPU-only (baseline)	Intel (48)	–	192 GB
CPU-only (HBM)	Intel (112)	–	128 GB (HBM)
CPU-only (large memory)	Intel (80)	–	4 TB (NVDIMM)

J.4 Ablation-study hyperparameters

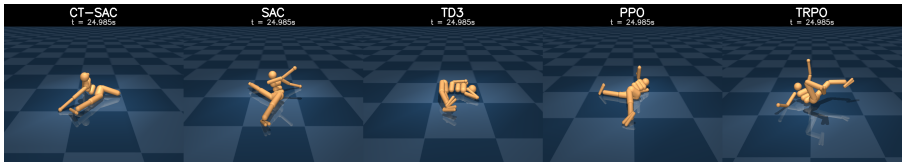
For completeness, we list the hyperparameter configurations used in the ablation study. Table 7, 8, 9, and 10 provide the configuration dictionary for each algorithm group, with **top/second/third** referring to the ranked settings used in the ablation tables. Due to space limit, we only list differing hyperparameters for these configurations. Please refer to the codebase to get a complete set of hyperparameters.



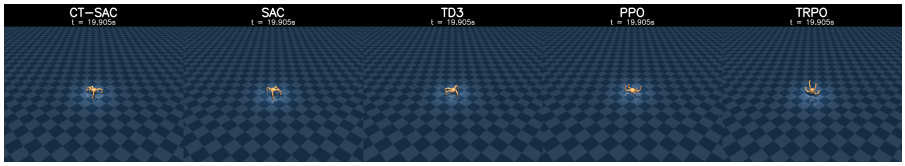
(a) Cheetah



(b) Walker



(c) Humanoid



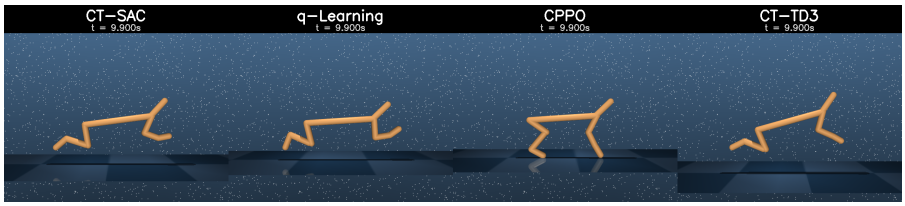
(d) Quadruped

Figure 9: CT-SAC vs. discrete-time baselines under irregular time steps.

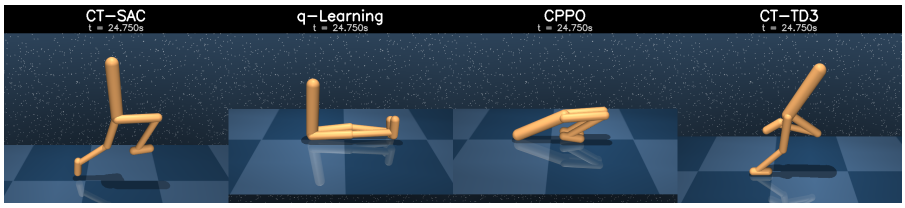
Table 7: Hyperparameter dictionary (CT-SAC vs SAC). For each environment, **top/second/third** correspond to the ranked configurations used in Table 2.

Task	**CT-SAC**			**SAC**		
	top	**second**	**third**	**top**	**second**	**third**
Cheetah	lr=0.00073, tau=0.04	lr=0.00073, tau=0.02	lr=0.00146, tau=0.04	lr=0.00073, train_freq=4, gradient_steps=4	lr=0.00073, train_freq=1, gradient_steps=1	lr=0.00146, train_freq=4, gradient_steps=4
Walker	lr=0.00073, train_freq=1, gradient_steps=1, tau=0.02	lr=0.00073, train_freq=4, gradient_steps=4, tau=0.02	lr=0.00146, train_freq=4, gradient_steps=4, tau=0.01	lr=0.00146, train_freq=1, gradient_steps=1	lr=0.00073, train_freq=4, gradient_steps=4	lr=0.00292, train_freq=1, gradient_steps=1
Humanoid	train_freq=1, gradient_steps=1, tau=0.01	train_freq=4, gradient_steps=4, tau=0.005	train_freq=1, gradient_steps=1, tau=0.005	log_std_init=-0.5, lr=0.0003, tau=0.0025	log_std_init=0.5, lr=0.0003, tau=0.005	log_std_init=-0.5, lr=0.0012, tau=0.005
Quadruped	train_freq=4, gradient_steps=4, tau=0.02	train_freq=1, gradient_steps=1, tau=0.04	train_freq=1, gradient_steps=1, tau=0.02	lr=0.000182	lr=0.00292	lr=0.00146
Trading	lr=0.001, train_freq=4, gradient_steps=4, tau=0.02	lr=0.0005, train_freq=1, gradient_steps=1, tau=0.02	lr=0.001, train_freq=1, gradient_steps=1, tau=0.005	log_std_init=-3.0, lr=0.002, train_freq=1, gradient_steps=1	log_std_init=-1.0, lr=0.002, train_freq=4, gradient_steps=4	log_std_init=-3.0, lr=0.001, train_freq=1, gradient_steps=1

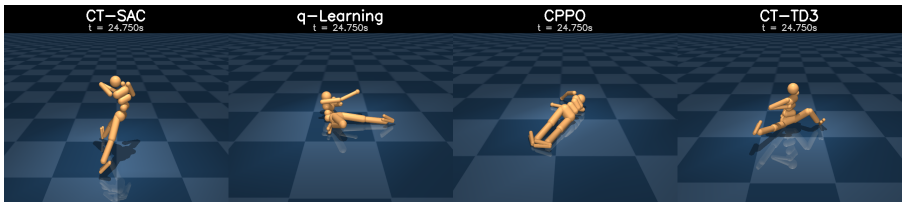
Abbreviation: lr=learning_rate; tau=soft-update rate; train_freq=update frequency; gradient_steps=number of gradient steps per update; log_std_init=initial policy log-std.



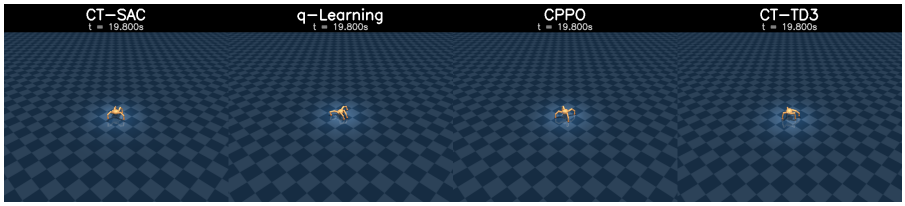
(a) Cheetah



(b) Walker



(c) Humanoid



(d) Quadruped

Figure 10: CT-SAC vs. continuous-time baselines under **regular** time steps. Note that the model is trained under irregular time settings.

K Benchmarking algorithms

K.1 Coupled-style q-learning

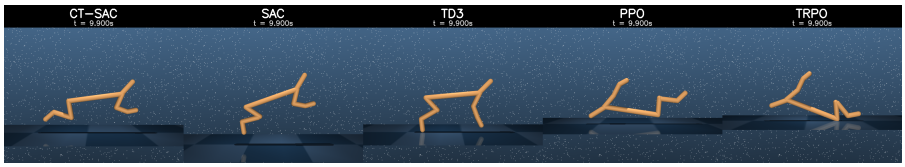
The instantaneous q-function introduced in Section 2.3 resolves the degeneracy of the discrete-time Q-function in the infinitesimal limit. A second obstacle remains: under diffusion dynamics, the usual squared TD regression is no longer aligned with the continuous-time dynamic programming principle. Consider the TD error with step size $u > 0$,

$$\delta_u \approx e^{-\beta u} V(X_{t+u}) + u\, r(X_t, a_t) - V(X_t) \tag{165}$$

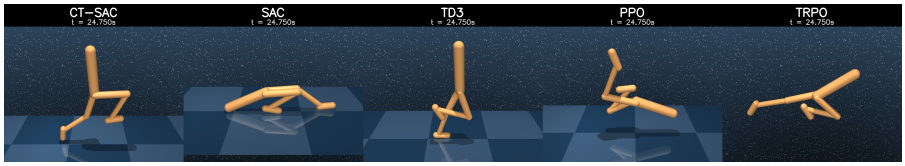
In a discrete-time MDP, minimizing $\mathbb{E}[\delta_u^2]$ enforces Bellman consistency. In continuous time, however, the discounted value increment contains not only a drift term but also a stochastic martingale term. In particular, with Itô lemma, rewriting δ_u in differential form as $u \to 0$ yields:

$$\begin{aligned} &d\left(e^{-\beta t} V(X_t)\right) + e^{-\beta t} r(X_t, a_t) dt \\ &= e^{-\beta t}\Big((L^{a_t} V)(X_t) - \beta V(X_t) + r(X_t, a_t) \Big)\, dt + e^{-\beta t} \nabla V(X_t)\, \sigma(X_t, a_t)\, dW_t \end{aligned} \tag{166}$$

As $u \to 0$, these two terms scale differently: the drift is $O(u)$, while the stochastic increment is $O(\sqrt{u})$. Hence, δ_u^2 is dominated by martingale variation rather than the Bellman drift. To address



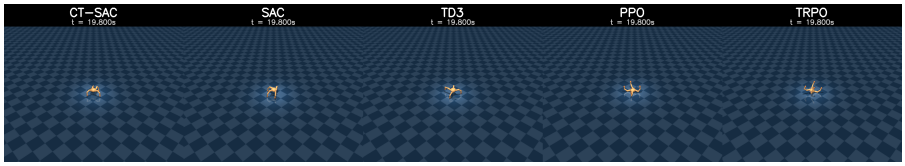
(a) Cheetah



(b) Walker



(c) Humanoid



(d) Quadruped

Figure 11: CT-SAC vs. discrete-time baselines under **regular** time steps. Note that the model is trained under irregular time settings.

Table 8: Hyperparameter dictionary (CT-TD3 vs TD3). For each environment, **top/second/third** correspond to the ranked configurations used in Table 2.

Task	**CT-TD3**			**TD3**		
	top	**second**	**third**	**top**	**second**	**third**
Cheetah	noise_sigma=0.1, policy_noise=0.4 noise_clip=1.0	noise_sigma=0.2, policy_noise=0.2 noise_clip=0.5	noise_sigma=0.1, policy_noise=0.2 noise_clip=0.3	lr=0.00073, policy_noise=0.3 noise_clip=0.6	lr=0.00073, policy_noise=0.1 noise_clip=0.2	lr=0.00146, policy_noise=0.1, noise_clip=0.2
Walker	lr=0.000365, policy_noise=0.2 noise_clip=0.5	lr=0.00073, policy_noise=0.2 noise_clip=0.5	lr=0.000365, policy_noise=0.1 noise_clip=0.2	lr=0.001, policy_noise=0.1 noise_clip=0.2	lr=0.0006, policy_noise=0.1 noise_clip=0.2	lr=0.0003, policy_noise=0.1, noise_clip=0.2
Humanoid	policy_noise=0.2 noise_clip=1.0	policy_noise=0.2 noise_clip=0.5	policy_noise=0.1 noise_clip=0.2	lr=0.0003, policy_noise=0.1 noise_clip=0.2	lr=0.0003, policy_noise=0.2 noise_clip=0.5	lr=0.0006, policy_noise=0.2, noise_clip=0.5
Quadruped	noise_sigma=0.1, policy_noise=0.4 noise_clip=1.0	noise_sigma=0.1, policy_noise=0.2 noise_clip=0.5	noise_sigma=0.2, policy_noise=0.2 noise_clip=0.5	lr=0.00073, policy_noise=0.4 noise_clip=1.0	lr=0.0003, policy_noise=0.2 noise_clip=0.5	lr=0.00073, policy_noise=0.2, noise_clip=0.5
Trading	noise_sigma=0.1, policy_noise=0.4 noise_clip=1.0	noise_sigma=0.1, policy_noise=0.1 noise_clip=0.2	noise_sigma=0.2, policy_noise=0.2 noise_clip=0.5	lr=0.001, policy_noise=0.2 noise_clip=0.3	lr=0.002, policy_noise=0.2 noise_clip=0.5	lr=0.001, policy_noise=0.3, noise_clip=0.6

Abbreviation: lr=learning_rate; policy_noise=noise std for target action; noise_clip=clip range for that noise; noise_sigma=exploration/target noise scale.

this mismatch, Jia & Zhou [11] characterize optimality via a martingale condition: for any start time t_0, the process:

$$M_t := e^{-\beta t} V(X_t) + \int_{t_0}^{t} e^{-\beta s} \big(r(X_s, a_s) - q(X_s, a_s) \big)\, ds$$

Table 9: Hyperparameter dictionary (CPPO vs PPO). For each environment, **top/second/third** correspond to the ranked configurations used in Table 2.

Task	CPPO			PPO		
	top	second	third	top	second	third
Cheetah	lr=0.00073, alpha=0.004, clip_ratio=False, clip_range=0.2, kl_coef_init=1.0	lr=0.00073, alpha=0.0004, clip_ratio=True, clip_range=0.1, kl_coef_init=0.3	lr=0.00146, alpha=0.0004, clip_ratio=False, clip_range=0.2, kl_coef_init=1.0	lr=0.000365, clip_range=0.2, vf_coef=0.5, gae_lambda=0.95	lr=0.000365, clip_range=0.1, vf_coef=0.5, gae_lambda=0.95	lr=0.000365, clip_range=0.2, vf_coef=1.0, gae_lambda=0.8
Walker	lr=0.00073, alpha=0.004, clip_ratio=False, clip_range=0.2, kl_coef_init=1.0	lr=0.000365, alpha=0.004, clip_ratio=False, clip_range=0.2, kl_coef_init=1.0	lr=0.00146, alpha=0.004, clip_ratio=False, clip_range=0.2, kl_coef_init=1.0	lr=0.000365, clip_range=0.2, vf_coef=1.0, gae_lambda=0.95	lr=0.000365, clip_range=0.2, vf_coef=0.5, gae_lambda=0.95	lr=0.000365, clip_range=0.2, vf_coef=0.5, gae_lambda=0.8
Humanoid	log_std_init=-0.5, alpha=0.004	log_std_init=-1.0, alpha=0.004	log_std_init=-1.0, alpha=0.0004	lr=0.0003, clip_range=0.2, vf_coef=0.5	lr=0.0003, clip_range=0.3, vf_coef=0.5	lr=0.0006, clip_range=0.2, vf_coef=0.5
Quadruped	log_std_init=-3.0, clip_ratio=False, clip_range=0.3	log_std_init=-3.0, clip_ratio=False, clip_range=0.2	log_std_init=-1.0, clip_ratio=False, clip_range=0.2	lr=0.000365, clip_range=0.2, vf_coef=0.5, gae_lambda=0.95	lr=0.000182, clip_range=0.2, vf_coef=0.5, gae_lambda=0.95	lr=0.000365, clip_range=0.2, vf_coef=0.5, gae_lambda=0.8
Trading	alpha=0.0004, sqrt_kl=0.02	alpha=0.04, sqrt_kl=0.02	alpha=0.0004, sqrt_kl=0.01	lr=0.002, clip_range=0.2, clip_range_vf=0.2, vf_coef=0.5	lr=0.001, clip_range=0.3, clip_range_vf=0.2, vf_coef=0.5	lr=0.001, clip_range=0.2, clip_range_vf=0.2, vf_coef=1.0

Abbreviation: lr=learning_rate; alpha=entropy coefficient; clip_range=PPO clipping range; clip_range_vf=value-function clipping; vf_coef=value loss coefficient; gae_lambda=GAE λ; log_std_init=initial policy log-std; sqrt_kl=target $\sqrt{\mathrm{KL}}$ constraint (CPPO).

Table 10: Hyperparameter dictionary (TRPO vs q-learning). For each environment, **top/second/third** correspond to the ranked configurations used in Table 2.

Task	TRPO			q-learning		
	top	second	third	top	second	third
Cheetah	lr=0.000365, gae_lambda=0.95, cg_damping=0.1	lr=0.000182, gae_lambda=0.95, cg_damping=0.1	lr=0.000365, gae_lambda=0.8, cg_damping=0.1	train_freq=1, gradient_steps=1, max_grad_norm=0.5	train_freq=4, gradient_steps=4, max_grad_norm=0.5	train_freq=4, gradient_steps=4, max_grad_norm=1.0
Walker	lr=0.000365, gae_lambda=0.95, cg_damping=0.1	lr=0.000182, gae_lambda=0.95, cg_damping=0.1	lr=0.000365, gae_lambda=0.8, cg_damping=0.1	log_std_init=-3, lr=0.000365	log_std_init=-1, lr=0.000365	log_std_init=-3, lr=0.000182
Humanoid	lr=0.0006, target_kl=0.01	lr=0.0003, target_kl=0.01	lr=0.001, target_kl=0.01	log_std_init=-1, lr=0.000365	log_std_init=-3, lr=0.000365	log_std_init=-1, lr=0.000182
Quadruped	lr=0.000365, gae_lambda=0.95, cg_damping=0.1	lr=0.000182, gae_lambda=0.8, cg_damping=0.1	lr=0.000365, gae_lambda=0.8, cg_damping=0.1	lr=0.000365, log_std_init=-3	lr=0.00073, log_std_init=-3	lr=0.00146, log_std_init=-3
Trading	lr=0.001, target_kl=0.005	lr=0.002, target_kl=0.01	lr=0.0005, target_kl=0.01	lr=0.004, max_grad_norm=1.0, log_std_init=-3	lr=0.002, max_grad_norm=0.5, log_std_init=-3	lr=0.001, max_grad_norm=1.0, log_std_init=-1

Abbreviation: lr=learning_rate; gae_lambda=GAE λ; target_kl=TRPO KL constraint; cg_damping=conjugate-gradient damping; max_grad_norm=gradient clipping threshold; train_freq=update frequency; gradient_steps=number of gradient steps per update; log_std_init=initial policy log-std.

must be a martingale. Two practical relaxations follow from this characterization: (i) a *martingale-loss* objective built from squared pathwise residuals (involving an outer integral over t and an inner integral over future s), and (ii) *martingale orthogonality* conditions of the form $\mathbb{E}\int_0^T \omega_t\big(dV(X_t) + ((r-q)-\beta V)\,dt\big) = 0$ for suitable test processes ω_t. To implement both criteria, we consider an (irregular) grid $t_0 < t_1 < \cdots < t_K = T$ with $\Delta_{t_i} := t_{i+1} - t_i$, and samples $(x_{t_i}, a_{t_i}, r_{t_i})$. Define the discounted martingale increment over $[t_k, T]$:

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

Two commonly used enforcement criteria are:

Algorithm 3 Coupled-style q-learning (Martingale loss)

Input: discount β, terminal time $T = t_K$, learning rates η_V, η_q.
Initialize parameters θ for value V_θ and ψ for advantage-rate q_ψ.
for $k_{\text{iter}} = 0, 1, 2, \ldots$ **do**
  Collect rollouts under a behavior policy, producing irregular grid samples $\{(x_{t_i}, a_{t_i}, r_{t_i}, \Delta_{t_i})\}_{i=0}^{K-1}$ and terminal payoff $h(x_{t_K})$.
  Sample a mini-batch of trajectories (or segments) from the dataset/buffer.
  for each trajectory in the mini-batch **do**
    for $k = 0, 1, \ldots, K-1$ **do**
      Compute

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

    end for
    Form loss $\mathcal{L}_{\text{mg}}(\theta, \psi) := \sum_{k=0}^{K-1} G_{t_k:T}(V_\theta, q_\psi)^2 \, \Delta_{t_k}$.
    Update critics (SGD): $\theta \leftarrow \theta - \eta_V \nabla_\theta \mathcal{L}_{\text{mg}}(\theta, \psi)$, $\psi \leftarrow \psi - \eta_q \nabla_\psi \mathcal{L}_{\text{mg}}(\theta, \psi)$.
  end for
end for

Algorithm 4 Coupled-style q-learning (Martingale orthogonality)

Input: discount β, learning rates η_V, η_q.
Initialize parameters θ for value V_θ and ψ for advantage-rate q_ψ.
for $k_{\text{iter}} = 0, 1, 2, \ldots$ **do**
  Collect transitions with irregular increments (x, a, r, x', Δ) and store in a buffer.
  Sample a mini-batch $\mathcal{B}$ of transitions from the buffer.
  for each (x, a, r, x', Δ) in $\mathcal{B}$ **do**
    Compute one-step residual

$$\delta := V_\theta(x') - V_\theta(x) + \big(r - q_\psi(x, a) - \beta V_\theta(x) \big) \Delta.$$

    Update value (test function $\omega_1 = \nabla_\theta V_\theta$): $\theta \leftarrow \theta + \eta_V \, \nabla_\theta V_\theta(x) \, \delta$.
    Update advantage-rate (test function $\omega_2 = \nabla_\psi q_\psi$): $\psi \leftarrow \psi + \eta_q \, \nabla_\psi q_\psi(x, a) \, \delta$.
  end for
end for

1. **Martingale loss (pathwise squared residual)** (See Algorithm 3):

$$\mathcal{L}_{\text{mg}}(V, q) := \sum_{k=0}^{K-1} G_{t_k:T}(V, q)^2 \, \Delta_{t_k}.$$

2. **Martingale orthogonality (test-function TD):** define the one-step residual

$$\delta_k := V(x_{t_{k+1}}) - V(x_{t_k}) + \big(r_{t_k} - q(x_{t_k}, a_{t_k}) - \beta V(x_{t_k}) \big) \Delta_{t_k},$$

   and enforce $\mathbb{E}[\omega(x_{t_k}, a_{t_k}) \, \delta_k] = 0$ for a class of test functions ω (value gradient in this case), yielding simple SGD-style updates (See Algorithm 4).

In Section 5, we used **Martingale orthogonality** method in the final report, as it yields better empirical performance. This variant is also used in [11] for their empirical evaluation.

K.2 Policy gradient

We denote a regularizer by $p(x, a; \pi)$ (for entropy regularization, $p(x, a; \pi) = -\log \pi(a|x)$) and the performance $\eta(\pi) := \int_{\mathcal{S}} V(x; \pi) \, \mu(dx)$. Below are two continuous-time policy gradient methods where CPPO (see Algorithm 6) is an upgraded version over the (vanilla) CPG (see Algorithm 5). For CPPO, due to its instability, we use 1.5 instead of 2 for penalty adaptation.

Algorithm 5 Continuous-time policy gradient (CPG)

1: **Input:** discount β, temperature α, learning rates η_V, η_π.
2: **Models:** value V_ϕ, policy π_θ.
3: **for** $k = 0, 1, 2, \dots$ **do**
4: Collect transitions (x, a, r, x', Δ) from rollouts under current policy π_θ, where $\Delta = t' - t$ can vary.
5: **Critic update (martingale orthogonality):**

$$\delta_V := V_\phi(x') - V_\phi(x) + \Delta\big(r(x,a) + \alpha\, p(x,a;\pi_\theta)\big) - \beta V_\phi(x)\Delta,$$

and perform SGD: $\phi \leftarrow \phi + \eta_V \nabla_\phi V_\phi(x)\, \delta_V$.
6: **Advantage-rate estimate (one-step / GAE-style):**

$$\hat{q}(x,a) := r + \frac{e^{-\beta\Delta} V_\phi(x') - V_\phi(x)}{\Delta}.$$

7: **Policy gradient step:** update θ using a mini-batch estimator of

$$\nabla_\theta \eta(\theta) \approx \frac{1}{\beta} \nabla_\theta \log \pi_\theta(a|x) \left(\hat{q}(x,a) + \alpha\, p(x,a;\pi_\theta)\right) \; + \; \frac{\alpha}{\beta} \nabla_\theta p(x,a;\pi_\theta).$$

8: $\theta \leftarrow \theta + \eta_\pi \widehat{\nabla_\theta \eta(\theta)}$.
9: **end for**

Algorithm 6 Continuous-time proximal policy optimization (CPPO, KL-penalty)

1: **Input:** discount β, temperature α, target KL δ_{KL}, tolerance ϵ, initial penalty C_{pen}, actor epochs E.
2: **Models:** value V_ϕ, policy π_θ.
3: **for** $k = 0, 1, 2, \dots$ **do**
4: Collect a batch $\mathcal{B} = \{(x, a, r, x', \Delta)\}$ from rollouts under current policy π_{θ_k}.
5: Update critic V_ϕ using the CPG martingale-orthogonality step (Algorithm 5, line 5).
6: Compute advantage-rate estimates $\hat{q}(x,a)$ on $\mathcal{B}$ (Algorithm 5, line 6) and (optionally) normalize.
7: Set $\theta \leftarrow \theta_k$.
8: **for** epoch $e = 1, \dots, E$ **do**
9: For mini-batches from $\mathcal{B}$, compute importance ratio $w_\theta(x,a) := \pi_\theta(a|x)/\pi_{\theta_k}(a|x)$.
10: Define penalized surrogate objective

$$J(\theta) := \mathbb{E}_{(x,a)\sim\mathcal{B}}\big[w_\theta(x,a)\big(\hat{q}(x,a) + \alpha\, p(x,a;\pi_\theta)\big)\big] - C_{\mathrm{pen}}\, \bar{D}_{\mathrm{KL}}(\theta\|\theta_k),$$

where $\bar{D}_{\mathrm{KL}}(\theta\|\theta_k) := \mathbb{E}_{x\sim\mathcal{B}}\Big[\sqrt{D_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|x)\,\|\,\pi_\theta(\cdot|x))}\Big]$.
11: Update actor by gradient ascent on $J(\theta)$.
12: **end for**
13: Set $\theta_{k+1} \leftarrow \theta$.
14: **Penalty adaptation:** if $\bar{D}_{\mathrm{KL}}(\theta_{k+1}\|\theta_k) \geq (1+\epsilon)\delta_{\mathrm{KL}}$, set $C_{\mathrm{pen}} \leftarrow 2C_{\mathrm{pen}}$; else if $\bar{D}_{\mathrm{KL}}(\theta_{k+1}\|\theta_k) \leq \delta_{\mathrm{KL}}/(1+\epsilon)$, set $C_{\mathrm{pen}} \leftarrow \frac{1}{2}C_{\mathrm{pen}}$.
15: **end for**

L LIMITATION

This work focuses on continuous state and action spaces; extending the framework to discrete or hybrid action spaces is an important direction for future work. Our theoretical analysis also relies on regularity assumptions on the value function, which help keep the resulting algorithm simple and scalable with strong empirical performance, while leaving room for sharper theory under weaker conditions. Importantly, our theory quantifies the approximation error at finite holding times u, rather than only in the asymptotic regime. As the holding time and algorithmic step size are refined, the asymptotic rates further show that this error vanishes under the conditions of our convergence results.

More broadly, continuous-time RL for irregular decision times remains underdeveloped, especially in scalable model-free settings. Our goal is to make this setting more practical while staying grounded in stochastic control. For this reason, we move beyond fixed-step simulated SDE benchmarks and evaluate on irregular-time control tasks and a real-world trading environment, which we view as closer to the challenges faced by practical RL systems.

The current appendices collect many ingredients needed to make the framework mathematically and experimentally complete, including theoretical assumptions, convergence arguments, implementation details, and reproducibility information. Each component can be further improved: tighter sample-complexity bounds, broader benchmark coverage, more robust implementations, and extensions to additional irregular-time domains are all promising directions.

Overall, we view irregular decision times as a natural generalization of standard discrete-time RL. Continuous-time RL is harder, but also more flexible, because it can naturally model non-uniform decision intervals. We hope this work helps make continuous-time RL more practical and opens the door to further development in this broader setting.