A Temporal Difference Method for Stochastic Continuous Dynamics

Haruki Settai **Naoya Takeishi** **Takehisa Yairi**
The University of Tokyo
{sharuki,ntake,yairi}@g.ecc.u-tokyo.ac.jp

Abstract

For continuous systems modeled by dynamical equations such as ODEs and SDEs, Bellman's principle of optimality takes the form of the Hamilton-Jacobi-Bellman (HJB) equation, which provides the theoretical target of reinforcement learning (RL). Although recent advances in RL successfully leverage this formulation, the existing methods typically assume the underlying dynamics are known a priori because they need explicit access to the coefficient functions of dynamical equations to update the value function following the HJB equation. We address this inherent limitation of HJB-based RL; we propose a model-free approach still targeting the HJB equation and propose the corresponding temporal difference method. We establish exponential convergence of the idealized continuous-time dynamics and empirically demonstrate its potential advantages over transition–kernel–based formulations. The proposed formulation paves the way toward bridging stochastic control and model-free reinforcement learning.

1 Introduction

Reinforcement learning (RL) has been successfully applied in various domains, ranging from discrete systems like board games (Silver et al., 2018) to systems that are continuous in both state and time such as robotic control (Kober et al., 2013). RL research has been advancing through a variety of approaches, and much of the work has focused on improving methods by refining objective functions (Schulman et al., 2015, 2017), balancing exploration and exploitation (Haarnoja et al., 2018), and developing more effective architectures (Hafner et al., 2019). These efforts have significantly advanced the field, leading to the development of various successful algorithms.

In contrast to the studies primarily targeting algorithmic components or optimization techniques, we focus on the continuity of time and explore what we call continuous RL, where the dynamics of systems are described by ordinary differential equations (ODEs) or stochastic differential equations (SDEs), which is a relatively underexplored aspect of RL. Although many RL methods have been applied, sometimes heuristically, to both discrete and continuous systems, the continuity of the system has not necessarily been fully exploited, even when it is known in advance. Laying a foundation for utilizing prior knowledge of continuity is important toward more effective learning and decision-making methods, particularly for practical applications such as robot control and autonomous driving, which typically fall into the target of continuous RL.

A way to incorporate prior knowledge of the system's continuity into the learning process is to use the Hamilton-Jacobi-Bellman (HJB) equation. In the Bellman equation, continuity is encoded in the transition kernel. However, in model-free RL, where transitions are approximated using samples, this continuity information is lost because the transition kernel is not explicitly modeled. On the other hand, the HJB equation retains the continuity information in the argument of the expectation rather than in the transition kernel. This property allows the HJB equation to keep taking continuity into

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

account even under sample-based approximations. However, because the HJB equation depends on the coefficient functions of the system's dynamics, prior work has largely been limited to model-based approaches (Munos and Bourgine, 1997; Yıldız et al., 2021).

In this paper, we introduce a temporal difference (TD) method based on the HJB equation, namely *differential TD* (dTD), achieved through sample-based approximation of the expectation term in the HJB equation. dTD enables policy evaluation without requiring knowledge or estimation of the system dynamics, while incorporating the continuity of the dynamics into the learning process. It is compatible with on-policy methods such as A2C (Mnih et al., 2016) and PPO (Schulman et al., 2017), and we demonstrate its effectiveness on Mujoco (Todorov et al., 2012) tasks including Hopper, HalfCheetah, Ant, and Humanoid. The codes for the proposed method are available at `https://github.com/4thhia/differential_TD`.

2 Related Work

Deterministic Dynamics The study of continuous RL for ODE systems can be traced back to studies such as Baird (1994); Munos (1997); Doya (2000); Munos (2006). Baird (1994) discovered that the Q-function collapses in continuous RL, which was rigorously proven and extended to deep RL in Tallec et al. (2019). In Munos (1997), model-free approaches for ODE systems were first studied. Doya (2000) was the first to introduce TD for ODE systems and extended it to TD(λ) and actor-critic. Munos (2006) investigated the estimation of policy gradients for ODE systems and proposed a pathwise derivative approach. More recently, Vamvoudakis and Crofton (2017) developed a model-free RL framework for deterministic linear systems. Kim et al. (2021) proposed a model-free Q-learning approach in which the control is derived from the HJB equation, while the learning target is based on the conventional Bellman equation. Yıldız et al. (2021) introduced a model-based method that leverages the Neural ODE framework to enable continuous-time optimization using learned system dynamics.

Stochastic Dynamics One of the earliest works on RL in SDE systems is Munos and Bourgine (1997), which takes a model-based approach. However, research in this direction remained largely unexplored until recently. In the past few years, a growing body of work has emerged that aims to establish theoretical foundations for RL in stochastic dynamics. Wang et al. (2020) introduced an entropy-regularized relaxed control formulation and provided a comprehensive analysis in the LQR setting. Tang et al. (2022) further demonstrated the well-posedness of the HJB equation within this relaxed control framework. Jia and Zhou (2022b) showed that Bellman optimality is equivalent to maintaining the martingale property of a suitably defined stochastic process, and proposed a corresponding algorithm. However, their method requires access to full trajectory information and thus applies only to finite-horizon settings. Building on this approach, Jia and Zhou (2022a, 2023) proposed actor-critic and Q-learning algorithms for finite-horizon SDE systems, respectively. Zhao et al. (2020) extended key theoretical tools such as the state visitation distribution and the performance difference lemma to the continuous-time setting and applied them to TRPO and PPO. Independently, Kobeissi and Bach (2023) proposed a variance-reduction technique for policy evaluation.

While HJB-based formulations have been extensively studied, existing model-free approaches inevitably resort to transition kernel representations, which obscure the information that the dynamics are continuous. We instead propose a model-free method grounded in stochastic dynamics that remains computationally tractable across broader tasks.

3 Background

3.1 Problem Setting

We consider a continuous RL setting where the state space is $\mathcal{S} \subset \mathbb{R}^n$ and the action space is $\mathcal{A}$. We suppose that the dynamics of the state are governed by the following controlled SDE:

$$dS_t = \mu(S_t, A_t)dt + \sigma(S_t, A_t)dB_t, \tag{1}$$

where $\mu : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^n$, $\sigma : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{n\times m}$, and $(B_t)_{t\geq 0}$ is the m-dimensional Brownian motion. Note that the state evolution is influenced by both the inherent noise in the system as well as the randomness induced by the stochastic policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, where $\mathcal{P}(\mathcal{A})$ is the space of probability distribution over the action space. Thus, the expectation related to this SDE is expressed as $\mathbb{E}_{p_\pi}[\cdot]$, where p_π denotes the transition probability density function corresponding to this SDE. For simplicity, we assume that the stochastic process (1) is well-defined; see Appendix A.1 for a detailed justification. We here focus on SDE systems because we can recover results for ODE systems in the limit of $\sigma = 0$.

3.2 Continuous RL

The goal of continuous RL is to find the policy π^* that maximizes the infinite horizon expected discounted cumulative reward:

$$\pi^*(s) = \operatorname*{argmax}_{\pi} \mathbb{E}_{p_\pi}\left[\int_0^\infty \mathrm{e}^{-\gamma t}\rho(S_t, A_t)dt \,\middle|\, S_0 = s\right],$$

where $\rho : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward rate function, and $\gamma \in (0, \infty)$ is a constant discount factor. Throughout the paper we assume that ρ is continuous in its respective arguments and bounded, an assumption used in establishing the well-posedness of the associated HJB equation (see Appendix A.1).

A typical approach for finding π^* is to use the optimal value function, which is defined as follows:

$$V^*(s) := \max_{\pi} \mathbb{E}_{p_\pi}\left[\int_t^\infty \mathrm{e}^{-\gamma(\tau - t)}\rho(S_\tau, A_\tau)d\tau \,\middle|\, S_t = s\right].$$

In a continuous-time MDP, π^* can be found by solving the Bellman equation:

$$V^*(s_t) = \max_{\pi} \mathbb{E}_{p_\pi}\left[\rho(s_t, A_t)\Delta t + \mathrm{e}^{-\gamma \Delta t}V^*(S_{t+\Delta t})\right], \tag{2}$$

where Δt is a small time interval and need not be constant. Similarly to standard discrete RL, this problem can be solved in a model-free setting by approximating the expectation on the right-hand side with samples and performing value iteration. However, in (2), the fact that the system follows an SDE is encoded only in the transition probability density (i.e., the subscript of the expectation), so dropping it by sample-based approximation disregards the prior knowledge of continuity and thus does not necessarily leverage the information.

3.3 HJB equation

Since an agent depends only on observations and update rules, a natural way to inform the agent that the system follows an SDE is to incorporate the SDE into the update rules. This can be achieved by further transforming (2) using the SDE (i.e., expanding $V(S_{t+\Delta t})$ via the Ito formula), resulting in the well-known HJB equation (see Appendix A.2 for more detail):

$$\begin{aligned} V^*(s_t) = \frac{1}{\gamma} \max_{\pi} \ \mathbb{E}_{p_\pi}\Bigg[&\rho(s_t, A_t) + \sum_{i=1}^{n} \mu^i(s_t, A_t)\frac{\partial V^*(s)}{\partial s^i}\bigg|_{s=s_t} \\ &+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\sigma(s_t, A_t)\sigma^\top(s_t, A_t)]^{ij}\frac{\partial^2 V^*(s)}{\partial s^i \partial s^j}\bigg|_{s=s_t}\Bigg], \end{aligned} \tag{3}$$

where μ^i and $[\sigma\sigma^\top]^{ij}$ denote the i-th element of μ and the (i, j)-th element of $\sigma\sigma^\top$, respectively.

4 TD method for Continuous Systems

Now, are we all ready to implement model-free value iteration just by approximating the expectation on the right-hand side of (3) with samples? The answer is no because the argument of the expectation includes the coefficient functions of the SDE, μ and σ, making it impossible to directly approximate the expectation with samples. In this section, we first propose a way to derive a TD from the HJB equation and then analyze its convergence properties.

4.1 Deriving TD from the HJB equation

To utilize HJB for model-free RL, we need to transform the argument of the expectation on the right-hand side of equation (3) into a sample-based expression. For this purpose, while many previous studies employed value iteration-based methods using the optimal value function, we adopt a policy iteration-based approach based on the value function for a fixed policy:

$$V^{\pi}(s) := \mathbb{E}_{p_{\pi}}\left[\int_{t}^{\infty} \mathrm{e}^{-\gamma(\tau-t)}\rho(S_{\tau}, A_{\tau})d\tau \;\middle|\; S_t = s\right].$$

Later we will discuss this choice more. The HJB equation under a fixed policy is

$$V^{\pi}(s_t) = \frac{1}{\gamma}\mathbb{E}_{p_{\pi}}\left[\rho(s_t, A_t) + \sum_{i=1}^{n}\mu^{i}(s_t, A_t)\frac{\partial V^{\pi}(s)}{\partial s^{i}}\bigg|_{s_t} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\sigma(s_t, A_t)\sigma^{\top}(s_t, A_t)]^{ij}\frac{\partial^{2} V^{\pi}(s)}{\partial s^{i}\partial s^{j}}\bigg|_{s_t}\right]. \tag{4}$$

We now present our main theoretical result. It gives the foundation for our model-free formulation of temporal-difference learning based on the HJB equation. The idea is that the drift and diffusion terms in the HJB equation can be equivalently expressed using limits of sample-based finite differences.

Proposition 1. *When a stochastic process $(S_t)_{t\geq 0}$ follows the SDE in* (1)*, we have*

$$\mathbb{E}_{p_{\pi}}\left[\mu^{i}(s_t, A_t)\right] = \lim_{\Delta t\to 0}\mathbb{E}_{p_{\pi}}\left[\frac{S^{i}_{t+\Delta t} - s^{i}_{t}}{\Delta t}\right] \tag{5}$$

and

$$\mathbb{E}_{p_{\pi}}\left[[\sigma(s_t, A_t)\sigma^{\top}(s_t, A_t)]^{ij}\right] = \lim_{\Delta t\to 0}\mathbb{E}_{p_{\pi}}\left[\frac{(S^{i}_{t+\Delta t} - s^{i}_{t})(S^{j}_{t+\Delta t} - s^{j}_{t})}{\Delta t}\right]. \tag{6}$$

Proof. For the first claim, we expand the i-th component of the SDE (1) using the Ito formula:

$$S^{i}_{t+\Delta t} = s^{i}_{t} + \mu^{i}(s_t, A_t)\Delta t + \sum_{j=1}^{m}\sigma^{ij}(s_t, A_t)(B^{j}_{t+\Delta t} - B^{j}_{t}) + O(\Delta t^{\frac{3}{2}}). \tag{7}$$

Since $B^{j}_{t+\Delta t} - B^{j}_{t}$ follows a zero-mean Gaussian and is independent of the state and the action,

$$\mathbb{E}_{p_{\pi}}\left[\sum_{j=1}^{m}\sigma^{ij}(s_t, A_t)(B^{j}_{t+\Delta t} - B^{j}_{t})\right] = \sum_{j=1}^{m}\mathbb{E}_{\pi}\left[\sigma^{ij}(s_t, A_t)\right]\mathbb{E}\left[B^{j}_{t+\Delta t} - B^{j}_{t}\right] = 0.$$

By taking the expectation of both sides of (7) and then letting $\Delta t \to 0$, the terms in $O(\Delta t^{\frac{3}{2}})$ vanish, and we obtain (5). For the second part of the claim, we begin by considering the product:

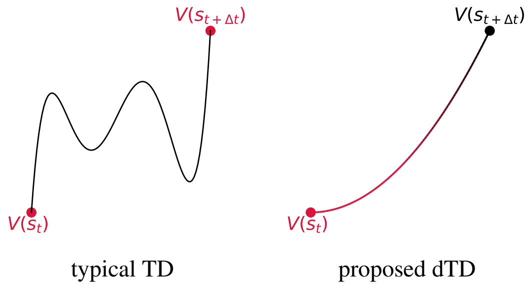
$$\begin{aligned}
&(S^{i}_{t+\Delta t} - s^{i}_{t})(S^{j}_{t+\Delta t} - s^{j}_{t}) \\
&= \mu^{i}(s_t, A_t)\mu^{j}(s_t, A_t)\Delta t^{2} + \sum_{k=1}^{m}\sum_{l=1}^{m}\sigma^{ik}(s_t, A_t)\sigma^{jl}(s_t, A_t)(B^{k}_{t+\Delta t} - B^{k}_{t})(B^{l}_{t+\Delta t} - B^{l}_{t}) \\
&\quad + \mu^{i}(s_t, A_t)\Delta t\sum_{k=1}^{m}\sigma^{jk}(s_t, A_t)(B^{k}_{t+\Delta t} - B^{k}_{t}) + \mu^{j}(s_t, A_t)\Delta t\sum_{k=1}^{m}\sigma^{ik}(s_t, A_t)(B^{k}_{t+\Delta t} - B^{k}_{t}) + O(\Delta t^{\frac{3}{2}}).
\end{aligned} \tag{8}$$

and then take the expectation of both sides. Using the fact

$$\mathbb{E}\left[(B^{k}_{t+\Delta t} - B^{k}_{t})(B^{l}_{t+\Delta t} - B^{l}_{t})\right] = \delta_{kl}\Delta t,$$

where $\delta_{kl} = 1$ if $k = l$ and $\delta_{kl} = 0$ otherwise, we can take the expectation of both sides of equation (8) and then let $\Delta t \to 0$, which yields (6). □

Remark 1. *Note that since $S_{t+\Delta t}$ in equations (5) and (6) is sampled under the policy π, our method is not applicable to off-policy settings such as value iteration or Q-learning. This limitation reflects a fundamental distinction: our formulation relies on the HJB equation under a fixed policy (i.e., policy evaluation), rather than the classical HJB equation involving maximization over all policies.*



typical TD

proposed dTD

Figure 1: Qualitative difference between the typical TD method and the proposed dTD method; the objects in red indicate what is adjusted by each temporal difference. (*Left*) In the typical TD method, the values of $\hat{V}$ are adjusted to minimize the TD error. (*Right*) In the dTD method, the gradient and the second derivative of $\hat{V}$ at s_t are adjusted to minimize the dTD error.

From Proposition 1, the HJB equation (4) can be reformulated as

$$V^\pi(s_t) = \frac{1}{\gamma} \lim_{\Delta t \to 0} \mathbb{E}_{p_\pi} \Bigg[\rho(s_t, A_t) + \sum_{i=1}^{n} \frac{S^i_{t+\Delta t} - s^i_t}{\Delta t} \frac{\partial V^\pi(s)}{\partial s^i}\bigg|_{s_t} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{(S^i_{t+\Delta t} - s^i_t)(S^j_{t+\Delta t} - s^j_t)}{\Delta t} \frac{\partial^2 V^\pi(s)}{\partial s^i \partial s^j}\bigg|_{s_t} \Bigg].$$

As we have rearranged the HJB equation so that the argument of expectation does not depend on the model, μ and σ, we can construct a temporal-difference update directly from it. We refer to this update as *differential temporal difference (dTD)* and expect it to be particularly effective when the observation interval Δt is small.

Definition 1 (differential temporal difference)**.** *Let* $\Delta t > 0$ *be a time step and* $\widehat{V}$ *denote an estimated value function. The dTD is defined as:*

$$\text{dTD} := \frac{1}{\gamma} \Bigg(\rho(s_t, a_t) + \sum_{i=1}^{n} \frac{s^i_{t+\Delta t} - s^i_t}{\Delta t} \frac{\partial \widehat{V}(s)}{\partial s^i}\bigg|_{s_t} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{(s^i_{t+\Delta t} - s^i_t)(s^j_{t+\Delta t} - s^j_t)}{\Delta t} \frac{\partial^2 \widehat{V}(s)}{\partial s^i \partial s^j}\bigg|_{s_t} \Bigg) - \widehat{V}(s_t). \tag{9}$$

As illustrated in Figure 1, unlike conventional TD methods based on transition kernels, dTD encourages the learning of a smooth value function by incorporating the continuity of the state space, even under sample-based approximation. We note that the version of dTD for ODE systems can be recovered by simply removing the term corresponding to the diffusion coefficient σ.

4.2 Convergence Analysis

To analyze the convergence, we first define the key operators for the fixed-policy HJB equation.

Definition 2 (HJB operator under a fixed policy)**.** *Given a stationary Markov policy* $\pi : S \to A$*, discount rate* $\gamma > 0$*, drift* $\mu : S \times A \to \mathbb{R}^n$*, diffusion* $\sigma : S \times A \to \mathbb{R}^{n \times m}$*, and instantaneous reward rate* $\rho : S \times A \to \mathbb{R}$*, the* HJB operator under a fixed policy T *maps any function* $V : S \to \mathbb{R}$ *to*

$$(TV)(s_t) := \frac{1}{\gamma} \mathbb{E}_{p_\pi} \Bigg[\rho(s_t, a_t) + \sum_{i=1}^{n} \mu^i(s_t, a_t) \frac{\partial V(s)}{\partial s^i}\bigg|_{s_t} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} [\sigma(s_t, a_t)\sigma^\top(s_t, a_t)]^{ij} \frac{\partial^2 V(s)}{\partial s^i \partial s^j}\bigg|_{s_t} \Bigg].$$

Definition 3 (Infinitesimal Generator)**.** *Given the same policy* π *and system dynamics* (μ, σ) *as in Definition 2, the* infinitesimal generator $\mathcal{L}^\pi$ *maps any* C^2 *function* $V : S \to \mathbb{R}$ *to*

$$(\mathcal{L}^\pi V)(s) := \mathbb{E}_{p_\pi} \Bigg[\sum_{i=1}^{n} \mu^i(s, A) \frac{\partial V(s)}{\partial s^i} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} [\sigma(s, A)\sigma^\top(s, A)]^{ij} \frac{\partial^2 V(s)}{\partial s^i \partial s^j} \Bigg].$$

Unlike the Bellman operator, whose convergence is guaranteed by its contraction property, the HJB operator involves unbounded differential operators and is generally not a contraction. Therefore, we establish convergence of the iterative scheme, i.e.,

$$V_{k+1} = V_k + \eta_k (TV_k - V_k)$$

by analyzing its continuous-time limit,

$$\frac{V_{k+1} - V_k}{\eta_k} = TV_k - V_k \overset{\eta_k \to 0}{\Longrightarrow} \frac{\partial V(t)}{\partial t} = TV - V,$$

under the idealized setting that the function V itself can be directly manipulated and updated (rather than updated via a finite set of parameters). The convergence analysis thus relies on showing that these dynamics asymptotically stabilize to the unique fixed point V satisfying $TV = V$. Since we focus on the HJB equation for a fixed policy, which lacks the $\max$ operator, this problem reduces to the analysis of a linear elliptic PDE. Using Definitions 2 and 3, and defining the expected reward rate as $\bar{\rho}(s) := \mathbb{E}_{\pi(\cdot|s)}[\rho(s, A)]$, the equation $V = TV$ is equivalent to:

$$(\gamma I - \mathcal{L}^\pi) V(s) = \bar{\rho}(s).$$

We show that the standard analysis for such PDEs, based on the Lax-Milgram theorem, can be applied to the bilinear form associated with the operator $(\gamma I - \mathcal{L}^\pi)$ to guarantee convergence to this unique fixed point. (We leave the analysis of errors from function approximation, which remains a challenging open problem even for standard Bellman operators, to future work.)

Assumption 1. *We assume the following conditions hold:*

1. ***(Domain)*** *The state space S is a bounded, connected open subset of $\mathbb{R}^n$ with a smooth boundary ∂S. We conduct our analysis in the Sobolev space $H^1(S)$.*
2. ***(Reflecting Boundary)*** *We assume the system satisfies a Neumann condition on ∂S:*
$$n(s) \cdot (D(s)\, \nabla V(s)) = 0 \quad \text{for } s \in \partial S,$$
where $D(s) = \mathbb{E}_{p_\pi}[\sigma(s, A)\sigma^\top(s, A)]$, and $n(s)$ is the outward unit normal on ∂S.
3. ***(Coefficients)*** *The policy-averaged reward $\bar{\rho}(s) := \mathbb{E}_{p_\pi}[\rho(s, A)]$, drift $\bar{\mu}(s) := \mathbb{E}_{p_\pi}[\mu(s, A)]$, and diffusion matrix $D(s)$ are bounded and Lipschitz continuous on $\bar{S}$.*
4. ***(Uniform Ellipticity)*** *The policy-averaged diffusion matrix $D(s)$ is symmetric and uniformly elliptic. That is, there exists a constant $\alpha > 0$ such that for all $s \in \bar{S}$ and $\xi \in \mathbb{R}^n$:*
$$\xi^\top D(s)\, \xi \geq \alpha \|\xi\|^2.$$
5. ***(Discount Factor)*** *The discount rate $\gamma > 0$ is sufficiently large to ensure the coercivity of the bilinear form associated with the operator $(\gamma I - \mathcal{L}^\pi)$.*

Using these assumptions, we first formally establish the existence and uniqueness of the solution V^π to the fixed-point equation $(\gamma I - \mathcal{L}^\pi)V = \bar{\rho}$. This solution V^π serves as the unique fixed point for our dynamics. A proof is deferred to Appendix A.3.

Lemma 1 (Existence and Uniqueness of the Fixed Point)**.** *Under Assumption 1, the linear elliptic PDE*

$$(\gamma I - \mathcal{L}^\pi) V(s) = \bar{\rho}(s) \quad \text{for } s \in S$$

admits a unique weak solution $V^\pi \in H^1(S)$.

Proposition 2 (Exponential Stability of the Dynamics)**.** *Let $V^\pi \in H^1(S)$ be the unique fixed point from Lemma 1. Under Assumption 1, the solution $V(t)$ of the continuous-time dynamics*

$$\frac{\partial V(t)}{\partial t} = TV(t) - V(t)$$

converges exponentially fast to V^π in the $L^2(S)$ norm. Specifically, there exists a constant $\lambda > 0$ such that for any initial condition $V(0) \in L^2(S)$:

$$\|V(t) - V^\pi\|_{L^2(S)} \leq e^{-\lambda t} \|V(0) - V^\pi\|_{L^2(S)}.$$

We comment on the interpretation and practicality of the assumptions in Assumption 1 as follows. **(1) Domain.** The state space S is bounded, which reflects physical or operational limits, such as joint limits or maximum velocity, making it a standard and natural modeling choice. **(2) Reflecting Boundary.** The Neumann condition ($n \cdot (D\nabla V) = 0$) models a reflecting boundary. This is physically natural for state-constrained systems that cannot exit S. **(3) Coefficients.** Bounded/Lipschitz $\bar{\mu}$, D, and $\bar{\rho}$ are classical regularity conditions ensuring well-posed SDEs and boundedness of the weak form; they are routinely satisfied by smooth dynamics and regular policies. **(4) Uniform ellipticity.** This implies the presence of non-degenerate random excitation (noise) in all directions. This is physically plausible for complex systems where environmental or internal noise can perturb the state in any dimension **(5) Discount.** The discount $\gamma > 0$ supplies a zero-order "anchor" that absorbs drift terms and yields a spectral gap; in practice, one can take γ large enough (with the usual horizon–myopia trade-off) to guarantee coercivity.

5 Method

This section outlines our method for applying dTD in deep reinforcement learning. Because dTD relies on function approximation, we restrict our attention to the deep RL regime, representing the value function with a neural network. A concise pseudocode listing is provided in Appendix B.1; here we explain the loss formulation and the β-dTD stabilization strategy.

5.1 Loss function

In deep RL, TD methods typically use a fixed target, known as the TD target, $r(s,t) + \gamma_{\text{discrete}} V(s_{t+1})$, as the teacher and aim to approximate the prediction $V(s_t)$ by minimizing the squared error between them. Although it may seem natural, by analogy with classical TD, to treat the $V(s_t)$ term in the dTD (9) as the prediction and regard the remaining terms as the dTD target, this is in fact unnecessary. As shown in Appendix A.2, the terms that appear in (4) and thus in (9) are derived through a series of transformations, and thus in dTD we no longer interpret the terms other than $V(s_t)$ as a low-variance estimate of the Bellman error. Consequently, the split between prediction and target is not unique.

We examine two different ways of defining the prediction and target from the rhs of (9).

- As a baseline, we first consider a naive formulation following the typical TD-style decomposition, that is, we treat $V(s_t)$ as the prediction. We refer to this approach as **naive-dTD**.
- On the other hand, inspired by the Taylor expansion of the Bellman equation, we treat the terms involving the derivative of $V(s_t)$ as the prediction and regard the rest as the target. We term such a more motivated parametrization simply as **dTD** hereafter.

These two variants are summarized in Table 1. As introduced in the next section, we empirically found that dTD performed significantly better than naive-dTD.

5.2 Hybrid scheme for stabilizing dTD

Although Proposition 2 establishes a convergence analysis, it relies on an idealized setting and does not cover the practical instability that can arise from function approximation errors in deep RL. Consequently, we cannot a priori guarantee that plain dTD operates stably and efficiently in practice.

To make the critic update more robust, we linearly combine the classical TD error with the dTD error, using weights $1 - \beta$ and β, respectively; we call the resulting update β-dTD. The TD part supplies the empirical stability that underpins most deep RL algorithms, whereas the dTD part injects gradient information from the continuous dynamics, accelerating learning when the underlying assumptions are approximately satisfied. We hypothesize that β-dTD can strike a balance to stabilize learning progress and potentially improve convergence behavior in practice.

5.3 Efficient Computation of the dTD Loss

Since equation (9) involves the Hessian, it may seem that $O(n^2)$ (where n is the dimension of the observation space) computations are required. However, by rearranging the order of calculations,

Table 1: Comparison of target and prediction terms in TD methods. Here, $\Delta s_t^i := s_{t+\Delta t}^i - s_t^i$ denotes the i-th component of the state transition over a small time interval Δt.

	Target	Prediction
TD	$r(s,t) + \gamma_{\text{discrete}} V(s_{t+1})$	$V(s_t)$
naive-dTD	$\rho(s_t, a_t) + \sum_{i=1}^{n} \frac{\Delta s_t^i}{\Delta t} \frac{\partial V(s)}{\partial s^i}\bigg\rvert_{s_t}$ $+\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\Delta s_t^i \Delta s_t^j}{\Delta t} \frac{\partial^2 V(s)}{\partial s^i \partial s^j}\bigg\rvert_{s_t}$	$\gamma V(s_t)$
dTD	$-\rho(s_t, a_t) + \gamma V(s_{t+\Delta t})$	$\sum_{i=1}^{n} \frac{\Delta s_t^i}{\Delta t} \frac{\partial V(s)}{\partial s^i}\bigg\rvert_{s_t}$ $+\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\Delta s_t^i \Delta s_t^j}{\Delta t} \frac{\partial^2 V(s)}{\partial s^i \partial s^j}\bigg\rvert_{s_t}$

such as using

$$\left\langle \Delta s_t,\ \frac{\partial^2 V(s)}{\partial s^2}\bigg|_{s_t} \Delta s_t \right\rangle = \left\langle \Delta s_t,\ \frac{\partial}{\partial s} \left\langle \frac{\partial V}{\partial s}, \Delta s_t \right\rangle \bigg|_{s_t} \right\rangle,$$

we can avoid directly calculating the Hessian and achieve a computation complexity of $O(n)$.

6 Experiments

6.1 Modification for discrete environment compatibility

In our theoretical framework, we work with continuous rewards (i.e., reward rate function) and a specific form of the discount ratio $e^{-\gamma}$, which is not directly compatible with the discrete discount ratio γ_{discrete}. To address this, we adjusted the reward and discount ratio following the same approach discussed in Tallec et al. (2019). The continuous reward formulation can be approximated by:

$$\int_0^\infty \mathrm{e}^{-\gamma t} \rho(s_t, a_t) dt \approx \sum_{k=0}^{\infty} \mathrm{e}^{(-\gamma \Delta t) k} \rho(s_{k\Delta t}, a_{k\Delta t}) \Delta t.$$

In this approximation, $\rho(s_t, a_t)\Delta t$ corresponds to the discrete reward r, and $e^{-\gamma \Delta t}$ corresponds to the discrete discount ratio γ_{discrete}. Thus, we can establish the following relationship:

$$\rho(s_t, a_t) = \frac{r(s_t, a_t)}{\Delta t} \quad \text{and} \quad \gamma = -\frac{1}{\Delta t} \log(\gamma_{\text{discrete}}).$$

This adjustment ensures that the observed discrete rewards are properly scaled to align with the continuous reward formulation used in the dTD method. With this scaling, dTD can be computed as

$$\text{dTD Target} : -r(s_t, a_t) - \log(\gamma_{\text{discrete}}) V(s_{t+\Delta t}) \quad \text{and}$$

$$\text{dTD Prediction} : \sum_{i=1}^{n} (s_{t+\Delta t}^i - s_t^i) \frac{\partial V(s)}{\partial s_i}\bigg|_{s_t} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (s_{t+\Delta t}^i - s_t^i)(s_{t+\Delta t}^j - s_t^j) \frac{\partial^2 V(s)}{\partial s_i \partial s_j}\bigg|_{s_t}.$$

6.2 Experiment design

Environment We conducted experiments with the Brax[1] library (Freeman et al., 2021) in the following environments: Hopper, HalfCheetah, Ant and Humanoid. Each environment provides a mid- to high-dimensional state space, with the number of state components varying across environments:

[1] `https://github.com/google/brax`

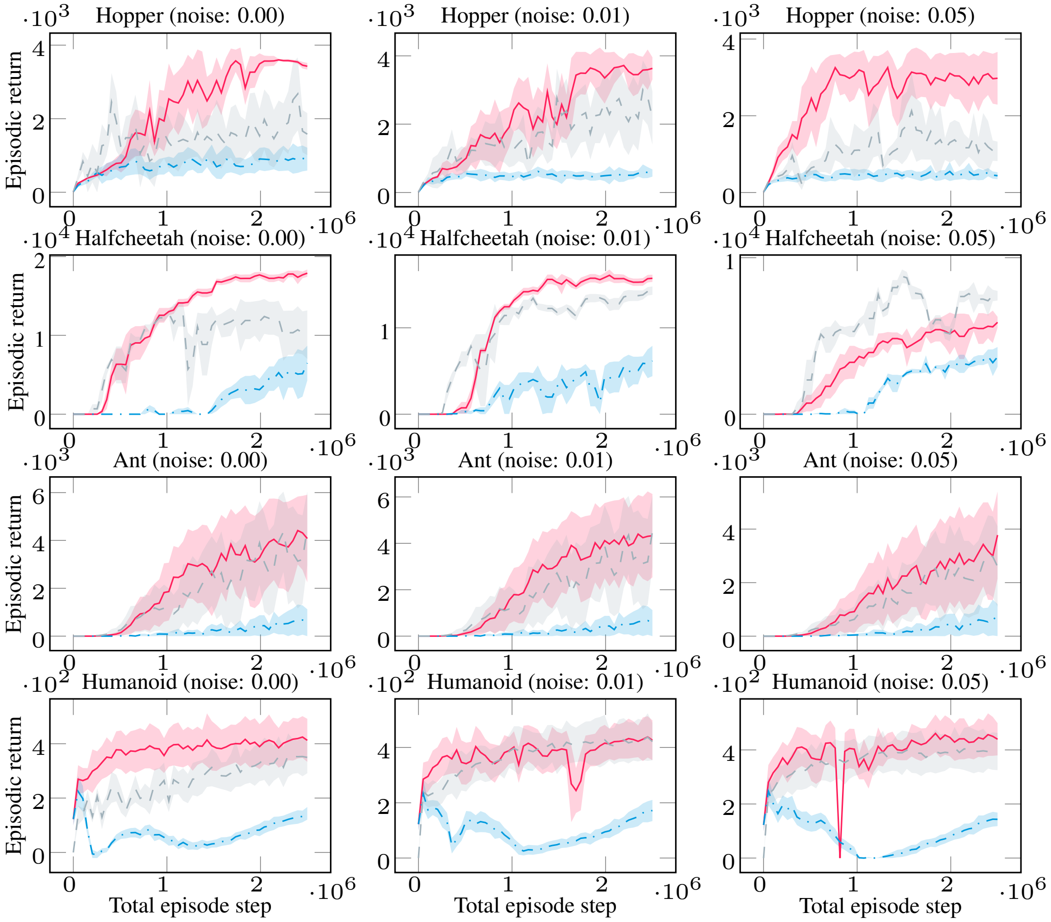


Figure 2: Performance of TD, β-naive-dTD, and β-dTD on continuous control benchmark. These results were obtained using the PPO algorithm. Each column corresponds to different noise levels (coef $= 0.00, 0.01, 0.05$), and each row corresponds to different environments. The tuned β values for β-naive-dTD were 0.08, 0.07, 0.23, 0.02 and for β-dTD were 0.57, 0.74, 0.24, 0.33 in Hopper, HalfCheetah, Ant, and Humanoid, respectively.

Hopper (11 dimensions), HalfCheetah (17 dimensions), Ant (27 dimensions) and Humanoid (244 dimensions). In each environment, at every step, we perturbed each state component by adding noise in the form of

$$s_i \leftarrow s_i + \text{coef} \times |s_i| \times \text{noise},$$

where noise $\sim \mathcal{N}(0, 1)$, and tested for three values of coef $= 0.00, 0.01, 0.05$. By adding this process noise, we aim to simulate with SDE systems, with the case of coef $= 0.00$ representing the limit case corresponding to ODEs. The specific time-step values used for each environment, which are not directly used in the learning process but are important to ensure they are small enough, are: Hopper: $\Delta t = 0.008$, HalfCheetah: $\Delta t = 0.05$, Ant: $\Delta t = 0.05$ and Humanoid: $\Delta t = 0.015$.

Baseline Various methods have been developed specifically for settings such as ODE (Tallec et al., 2019), LQR (Vamvoudakis and Crofton, 2017), or time-dependent Q functions with finite horizons (Jia and Zhou, 2023), but are often incompatible with the current deep RL framework (Kim et al., 2021) or rely on model-based assumptions (Munos and Bourgine, 1997), making them unsuitable for comparison with our proposed method. We chose to use standard TD methods as baselines and experimented with TD, β-naive-dTD, and β-dTD, using A2C(Mnih et al., 2016) and PPO (Schulman et al., 2017). In this paper, we present the results using PPO as the primary focus, and the results using A2C are provided in the Appendix B.3 for supplementary details.

Hyperparameter tuning For hyperparameter tuning, we applied the DEHB (Awad et al., 2021), a multi-fidelity method that is currently considered the most effective method in RL (Eimer et al., 2023). While we performed hyperparameter tuning for the standard PPO algorithm as well, we also reference the official tuning results from Freeman et al. (2021) for fair comparison. Additional details about the hyperparameter search space can be found in Appendix B.

6.3 Results and discussion

Comparative evaluation We compare (the variants of) the proposed method and the baseline:

(β-naive-dTD vs. β-dTD) In Figure 3, we can observe that β-dTD consistently outperforms β-naive-dTD. In all the environments, the optimized values of β for β-naive-dTD were quite small, suggesting that the effective update rule of β-naive-dTD became close to that of the standard TD. Despite such a fact, however, the performance of β-naive-dTD remains significantly worse than the standard TD. There are two possible explanations: (1) the β value chosen for β-naive-dTD was actually still not small enough to fully eliminate the adverse effect of the naive-dTD term; and (2) the TD-related parameters in β-naive-dTD were only suboptimally tuned because hyperparameter tuning resources were allocated mainly to optimizing β. These factors may jointly account for the unexpectedly poor performance of β-naive-dTD.

(TD vs. β-dTD) As shown in Figure 3, β-dTD outperforms TD or achieves comparable performance in all cases. While the degree of improvement varies, the final performance of TD and β-dTD tends to converge, which is not very surprising because both dTD and TD are derived from the same Bellman equation, and the resulting value functions should thus be similar to each other eventually. Nevertheless, dTD has the advantage of implicitly utilizing continuity information during training, which enables it to make more informative updates. Consequently, although the final performance may be comparable, β-dTD tends to show a faster rate of improvement relative to TD.

Significance of dTD In contrast to β-naive-dTD, the weight β in β-dTD is not exceedingly small. Notably, in the Halfcheetah environment, β assumes a relatively large value of 0.74. This indicates that dTD retains a meaningful impact on the learning process.

Impact of process noise In terms of robustness to process noise, both β-dTD and TD exhibit similar performance. When the noise level is coef $= 0.01$, neither method experiences significant degradation in performance. However, when the noise level is increased to coef $= 0.05$, both β-dTD and TD show similar reduction in performance, particularly in environments like Ant and Halfcheetah.

7 Conclusion

We have presented differential TD (dTD), a temporal difference method based on the HJB equation. In contrast to approaches based on transition kernels, the proposed method can incorporate the continuity of dynamics into the learning process without knowing the dynamics. We have shown empirical results for a variety of continuous control environments with different time intervals. The empirical results highlight the potential advantages of dTD in terms of learning speed and efficiency while also implying that stability concerns may exist in practice, which led to the introduction of the robust β-dTD update. Although the current paper focuses on the theoretical development of dTD, these observations are useful and also warrant further empirical exploration.

We have also analyzed the conditions under which the continuous-time dynamics of the HJB equation exhibit exponential stability toward the unique fixed point. This stability property, proven using techniques from linear elliptic PDE theory, is crucial for showing the theoretical convergence of the idealized iterative scheme. However, a drawback is that the sufficient conditions we identify, such as the requirement for a bounded domain and uniform ellipticity (Assumption 1), are often hard to maintain or verify in the context of deep RL with function approximation.

Future work includes bridging the gap between the theoretical exponential stability and the practical stability of dTD updates (e.g., by ensuring the coercivity condition in practice), reducing the variance of learning by improved estimators or regularization, and extending the wide range of existing TD-based techniques to the dTD framework.

Acknowledgements

NT was supported by JST PRESTO JPMJPR24T6, JSPS JP20K19869, and JSPS JP25H01454.

References

Noor Awad, Neeratyoy Mallik, and Frank Hutter. DEHB: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 2147–2153, 2021.

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

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

Theresa Eimer, Marius Lindauer, and Roberta Raileanu. Hyperparameters in reinforcement learning and how to tune them. In *Proceedings of the 40th International Conference on Machine Learning*, pages 9104–9149, 2023.

Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - a differentiable physics engine for large scale rigid body simulation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

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, pages 1861–1870, 2018.

Danijar Hafner, Timothy Lillicrap, and Jimmy Ba. Dream to control: Learning behaviors by latent imagination. arXiv: 1912.01603, 2019.

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), 2022a.

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), 2022b.

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

Jeongho Kim, Jaeuk Shin, and Insoon Yang. Hamilton-Jacobi deep Q-learning for deterministic continuous-time systems with Lipschitz continuous controls. *Journal of Machine Learning Research*, 22(206), 2021.

Ziad Kobeissi and Francis Bach. Temporal difference learning with continuous time and state in the stochastic setting. arXiv:1712.01815, 2023.

Jens Kober, J Andrew Bagnell, and Jan Peters. Some studies in machine learning using the game of checkers. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1928–1937, 2016.

Rémi Munos. A convergent reinforcement learning algorithm in the continuous case based on a finite difference method. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, volume 2, pages 826–831, 1997.

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

Rémi Munos and Paul Bourgine. Reinforcement learning for continuous stochastic control problems. In *Advances in Neural Information Processing Systems*, volume 10, 1997.

John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1889–1897, 2015.

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

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv:1712.01815, 2018.

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, pages 6096–6104, 2019.

Wenpin Tang, Paul Yuming Zhang, and Xun Yu Zhou. Exploratory hjb equations and their convergence. *SIAM Journal on Control and Optimization*, 60, 2022.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5026–5033. IEEE, 2012.

Kyriakos G. Vamvoudakis and Kevin T. Crofton. Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach. *Systems & Control Letters*, 100:14–20, 2017.

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), 2020.

Çağatay Yıldız, Markus Heinonen, and Harri Lähdesmäki. Continuous-time model-based reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12009–12018, 2021.

Hanyang Zhao, Wenpin Tang, and David D. Yao. Policy optimization for continuous reinforcement learning. In *Advances in Neural Information Processing Systems 36*, pages 13637–13663, 2020.

A Mathematical Details

A.1 Justification for the Continuous RL Formulation

In Section 3, we modeled the evolution of the state under a stochastic policy π by the controlled SDE

$$dS_t = \mu(S_t, A_t)\,dt + \sigma(S_t, A_t)\,dB_t, \quad A_t \sim \pi(\cdot|S_t).$$

Here, the control is applied in the form of action samples drawn from a stochastic policy at each time step. While this formulation closely reflects the sampling-based behavior in RL, it raises a technical challenge: the presence of external randomness in addition to the intrinsic Brownian noise introduces analytical difficulties. As a result, the well-definedness of this SDE is not immediately obvious.

To address this issue, many prior works (e.g., Wang et al. (2020); Jia and Zhou (2022b,a, 2023); Zhao et al. (2020)) adopt the averaged dynamics, denoted by $(\widetilde{S}_t)_{t\geq 0}$, whose distribution at each time t is known to coincide with that of the original one under the same initial condition (Wang et al., 2020). Specifically, the averaged dynamics is defined as

$$d\widetilde{S}_t = \widetilde{\mu}(\widetilde{S}_t, \pi)dt + \widetilde{\sigma}(\widetilde{S}_t, \pi)d\widetilde{B}_t,$$

where $\widetilde{\mu}(s,\pi) = \int_{\mathcal{A}} \mu(s,a)\pi(a)da,\ \widetilde{\sigma}(s,a) = \left(\int_{\mathcal{A}} \sigma(s,a)\sigma^\top(s,a)\pi(a)da\right)^{\frac{1}{2}}$ and $(\widetilde{B}_t)_{t\geq 0}$ is the m-dimensional Brownian motion. Since the averaged dynamics no longer involves the external randomness induced by stochastic action selection, its well-definedness is ensured by classical SDE theory under standard assumptions such as Lipschitz continuity and a linear growth condition.

Since the marginal distributions of the two dynamics coincide, the corresponding value functions also coincide:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_{p_\pi}\left[\int_t^\infty \mathrm{e}^{-\gamma(\tau-t)}\rho(S_\tau, A_\tau)\,d\tau \,\middle|\, S_t = s\right] \\ &= \mathbb{E}_{\widetilde{p}}\left[\int_t^\infty \mathrm{e}^{-\gamma(\tau-t)}\widetilde{\rho}(\widetilde{S}_\tau, \pi)\,d\tau \,\middle|\, \widetilde{S}_t = s\right] \\ &=: \widetilde{V}^\pi(s), \end{aligned}$$

where $\widetilde{\rho}(s,\pi) := \int_{\mathcal{A}} \rho(s,a)\,\pi(a)\,da$. Hence the value function above is itself well defined and raises no analytical issues.

A.2 Ito formula

The Bellman equation is given by:

$$V^*(s_t) = \max_\pi \mathbb{E}_{p_\pi}\left[\rho(s_t, A_t)\Delta t + \mathrm{e}^{-\gamma\Delta t}V^*(S_{t+\Delta t})\right].$$

Assuming that a stochastic process $(S_t)_{t\geq 0}$ follows the SDE (1), the term $V^*(S_{t+\Delta t})$ can be further expanded using Itô's lemma:

$$\begin{aligned} V^*(s_t) &= \max_{\pi}\ \mathbb{E}_{p_\pi}\Big[\rho(s_t,A_t)\Delta t + \mathrm{e}^{-\gamma\Delta t}V^*(S_{t+\Delta t})\Big] \\ &= \max_{\pi}\ \mathbb{E}_{p_\pi}\Bigg[\rho(s_t,A_t)\Delta t + \mathrm{e}^{-\gamma\Delta t}\Bigg\{V^*(s_t) + \Bigg(\sum_{i=1}^{n}\mu^i(s_t,A_t)\frac{\partial V^*(s)}{\partial s_i}\Bigg|_{s_t} \\ &\qquad + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\sigma(s_t,A_t)\sigma^\top(s_t,A_t)]^{ij}\frac{\partial^2 V^*(s)}{\partial s_i\partial s_j}\Bigg|_{s_t}\Bigg)\Delta t \\ &\qquad + \sum_{i=1}^{n}\sum_{j=1}^{m}\sigma^i_j(s_t,A_t)\frac{\partial V^*(s)}{\partial s_i}\Bigg|_{s_t}(B^j_{t+\Delta t}-B^j_t) + O((\Delta t)^{3/2})\Bigg\}\Bigg] \\ &= \max_{\pi}\ \mathbb{E}_{p_\pi}\Bigg[\rho(s_t,A_t)\Delta t + \mathrm{e}^{-\gamma\Delta t}\Bigg\{V^*(s_t) + \Bigg(\sum_{i=1}^{n}\mu^i(s_t,A_t)\frac{\partial V^*(s)}{\partial s_i}\Bigg|_{s_t} \\ &\qquad + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\sigma(s_t,A_t)\sigma^\top(s_t,A_t)]^{ij}\frac{\partial^2 V^*(s)}{\partial s_i\partial s_j}\Bigg|_{s_t}\Bigg)\Delta t + O((\Delta t)^{3/2})\Bigg\}\Bigg]. \end{aligned}$$

Simplifying the equation and taking the limit as $\Delta t \to 0$, we have the condition for the optimal value function:

$$\begin{aligned} V^*(s_t) = \frac{1}{\gamma}\max_{\pi}\ \mathbb{E}_{p_\pi}\Bigg[&\rho(s_t,A_t) + \sum_{i=1}^{n}\mu^i(s_t,A_t)\frac{\partial V^*(s)}{\partial s_i}\Bigg|_{s_t} \\ &+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\sigma(s_t,A_t)\sigma^\top(s_t,A_t)]^{ij}\frac{\partial^2 V^*(s)}{\partial s_i\partial s_j}\Bigg|_{s_t}\Bigg]. \end{aligned}$$

A.3 Convergence of dTD

A.3.1 Proof of Lemma 1

Proof. We consider the PDE $(\gamma I - \mathcal{L}^\pi)V = \bar{\rho}$ with

$$(\mathcal{L}^\pi V)(s) = \sum_{i=1}^{n}\bar{\mu}^i(s)\,\frac{\partial V}{\partial s^i} + \frac{1}{2}\sum_{i,j=1}^{n}D^{ij}(s)\,\frac{\partial^2 V}{\partial s^i\partial s^j}, \quad D(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[\sigma(s,a)\sigma^\top(s,a)].$$

Multiply by $v \in H^1(S)$, integrate over S, and integrate by parts once in s^i for the second–order term. Using the homogeneous Neumann boundary condition $n\cdot(D\nabla V) = 0$ on ∂S, we obtain

$$\int_S(\gamma V - \mathcal{L}^\pi V)\,v = \int_S \gamma V v + \tfrac{1}{2}\int_S\sum_{i,j}D^{ij}\,\partial_j V\,\partial_i v - \int_S\sum_i \bar{\mu}^i\,\partial_i V\,v - \tfrac{1}{2}\int_S\sum_{i,j}(\partial_i D^{ij})\,\partial_j V\,v.$$

Define

$$B(V,v) := \int_S \gamma V v + \tfrac{1}{2}\int_S\sum_{i,j}D^{ij}\,\partial_j V\,\partial_i v - \int_S\sum_i \bar{\mu}^i\,\partial_i V\,v - \tfrac{1}{2}\int_S\sum_{i,j}(\partial_i D^{ij})\,\partial_j V\,v, \quad f(v) := \int_S \bar{\rho}\,v.$$

Boundedness. By the bounded/Lipschitz coefficients in Assumption 1.3, there exists $C > 0$ such that

$$|B(V,v)| \leq C\,\|V\|_{H^1(S)}\,\|v\|_{H^1(S)}, \qquad |f(v)| \leq \|\bar{\rho}\|_{L^2(S)}\,\|v\|_{L^2(S)}.$$

Coercivity. Using uniform ellipticity (Assumption 1.4), we have

$$B(V,V) = \int_S \gamma V^2 + \tfrac{1}{2}\int_S(\nabla V)^\top D\,\nabla V - \int_S \bar{\mu}\cdot(\nabla V)\,V - \tfrac{1}{2}\int_S(\operatorname{div} D)\cdot(\nabla V)\,V.$$

Hence

$$B(V,V) \geq \gamma\|V\|^2_{L^2} + \frac{\alpha}{2}\|\nabla V\|^2_{L^2} - \Big(\|\bar{\mu}\|_\infty + \tfrac{1}{2}\|\operatorname{div} D\|_\infty\Big)\|\nabla V\|_{L^2}\|V\|_{L^2}.$$

By Young's inequality,

$$\Big(\|\bar{\mu}\|_\infty + \tfrac{1}{2}\|\mathrm{div}\, D\|_\infty\Big)\|\nabla V\|_{L^2}\|V\|_{L^2} \le \tfrac{\alpha}{4}\|\nabla V\|_{L^2}^2 + \frac{(\|\bar{\mu}\|_\infty + \frac{1}{2}\|\mathrm{div}\, D\|_\infty)^2}{\alpha}\|V\|_{L^2}^2.$$

Therefore

$$B(V,V) \ge \Big(\gamma - \frac{(\|\bar{\mu}\|_\infty + \frac{1}{2}\|\mathrm{div}\, D\|_\infty)^2}{\alpha}\Big)\|V\|_{L^2}^2 + \frac{\alpha}{4}\|\nabla V\|_{L^2}^2 \ge c\,\|V\|_{H^1(S)}^2$$

for some $c > 0$ ensured by Assumption 1.5.

Since $H^1(S)$ is a Hilbert space, B is bounded and coercive, and f is bounded, the Lax–Milgram Theorem implies there exists a unique weak solution $V^\pi \in H^1(S)$. □

A.3.2 Proof of Proposition 2

Proof. Set

$$J(t) := \frac{1}{2}\|V(t) - V^\pi\|_{L^2(S)}^2.$$

Since $TV = \frac{1}{\gamma}\big(\bar{\rho} + \mathcal{L}^\pi V\big)$ is affine and $TV^\pi = V^\pi$, we have

$$\frac{\partial}{\partial t}\big(V(t) - V^\pi\big) = TV(t) - V(t) - \big(TV^\pi - V^\pi\big) = -\gamma^{-1}\,(\gamma I - \mathcal{L}^\pi)\big(V(t) - V^\pi\big).$$

Differentiating J and using the bilinear form $B(\cdot,\cdot)$ from yields

$$\begin{aligned}\frac{dJ}{dt} &= \Big\langle V(t) - V^\pi,\ \frac{\partial}{\partial t}\big(V(t) - V^\pi\big)\Big\rangle \\ &= -\frac{1}{\gamma}\Big\langle V(t) - V^\pi,\ (\gamma I - \mathcal{L}^\pi)\big(V(t) - V^\pi\big)\Big\rangle \\ &= -\frac{1}{\gamma}\,B(V(t) - V^\pi,\ V(t) - V^\pi)\,.\end{aligned}$$

By coercivity of B, there exists $c > 0$ such that

$$B(V(t) - V^\pi,\ V(t) - V^\pi)\ \ge\ c\,\|V(t) - V^\pi\|_{H^1(S)}^2\ \ge\ 2c\,J(t).$$

Therefore,

$$\frac{dJ}{dt}\ \le\ -\frac{2c}{\gamma}\,J(t).$$

By Grönwall's inequality,

$$J(t)\ \le\ J(0)\,e^{-(2c/\gamma)t},$$

which implies the claimed exponential convergence in $L^2(S)$:

$$\|V(t) - V^\pi\|_{L^2(S)}\ \le e^{-(c/\gamma)t}\|V(0) - V^\pi\|_{L^2(S)}.$$

□

B Implementation Details

B.1 Algorithm

The procedures for policy evaluation are summarized in Algorithm 1.

Algorithm 1 Policy evaluation with dTD

```
Input: policy $\pi$
Output: $V_\theta$
  Initialize value function $V_\theta$ with random parameter $\theta$
  for each training step do
    Initialize buffer $\mathcal{D} = \emptyset$ and initial state $s_0$
    for each environment step do
      $a_t \sim \pi(\cdot|s_t)$
      $s_{t+\Delta t} \sim p(\cdot|s_t, a_t)$
      $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, s_{t+\Delta t}, \rho_t)$
    end for
    for each update step do
      Sample a batch of $D$ random transitions from $\mathcal{D}$
      $\bar{\theta} \leftarrow \theta$
      $y_d \leftarrow -\rho_t^d - \gamma V_{\bar{\theta}}(s_{t+\Delta t}^d)$
      pred$_d \leftarrow \sum_{i=1}^{n} \frac{(s_{t+\Delta t}^{d,i} - s_t^{d,i})}{\Delta t} \frac{\partial V_\theta(s)}{\partial s_i}\Big|_{s_t^d} + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{(s_{t+\Delta t}^{d,i} - s_t^{d,i})(s_{t+\Delta t}^{d,j} - s_t^{d,j})}{\Delta t} \frac{\partial^2 V_\theta(s)}{\partial s_i \partial s_j}\Big|_{s_t^d}$
      Update parameter $\theta$ using gradient descent method
      $\theta \leftarrow \mathrm{argmin}_\theta \frac{1}{D} \sum_{d=1}^{D} (y_d - \mathrm{pred}_d)^2$
    end for
  end for
```

B.2 Hyperparameters for PPO

The search space of the hyperparameters is summarized in Table 2. The values chosen finally are summarized in Tables **??**.

Table 2: Hyperparameter search space

Hyperparameter	Search Space
environment steps per update (number of parallel environment: 64)	$\{8, 16, 32\}$
number of epochs per update	range(5, 20)
minibatch size	$\{256, 512\}$
learning rate	$\log(\text{interval}(1e-6, 5e-3))$
normalize advantage	$\{\text{True}, \text{False}\}$
gae lambda	interval(0.8, 0.9999)
clip range	interval(0.0, 0.9)
entropy coefficient	interval(0.0, 0.3)
value loss weight	interval(0.0, 1.0)
mixture raio β	interval(0.0, 1.0)

Table 3: Best hyperparameters for PPO with TD and β-dTD across environments

Hyperparameter	TD				β-dTD			
	Hopper	Halfcheetah	Ant	Humanoid	Hopper	Halfcheetah	Ant	Humanoid
environment steps/update	32	16	8	16	32	8	32	16
epochs/update	7	5	11	15	19	9	16	10
minibatch size	512	256	512	512	256	256	256	256
learning rate	1.18e-3	5.93e-4	3.71e-4	1.40e-3	3.52e-4	3.29e-4	7.94e-5	1.08e-3
normalize advantage	False	False	False	False	False	False	False	False
GAE lambda	0.886	0.833	0.935	0.999	0.998	0.908	0.805	0.888
clip range	0.439	0.268	0.425	0.063	0.075	0.040	0.520	0.713
entropy coefficient	0.121	0.018	0.162	0.021	0.046	0.011	0.133	0.002
value loss weight	0.049	0.513	0.711	0.091	0.675	0.268	0.274	0.054
mixture ratio β	—	—	—	—	0.572	0.742	0.241	0.332

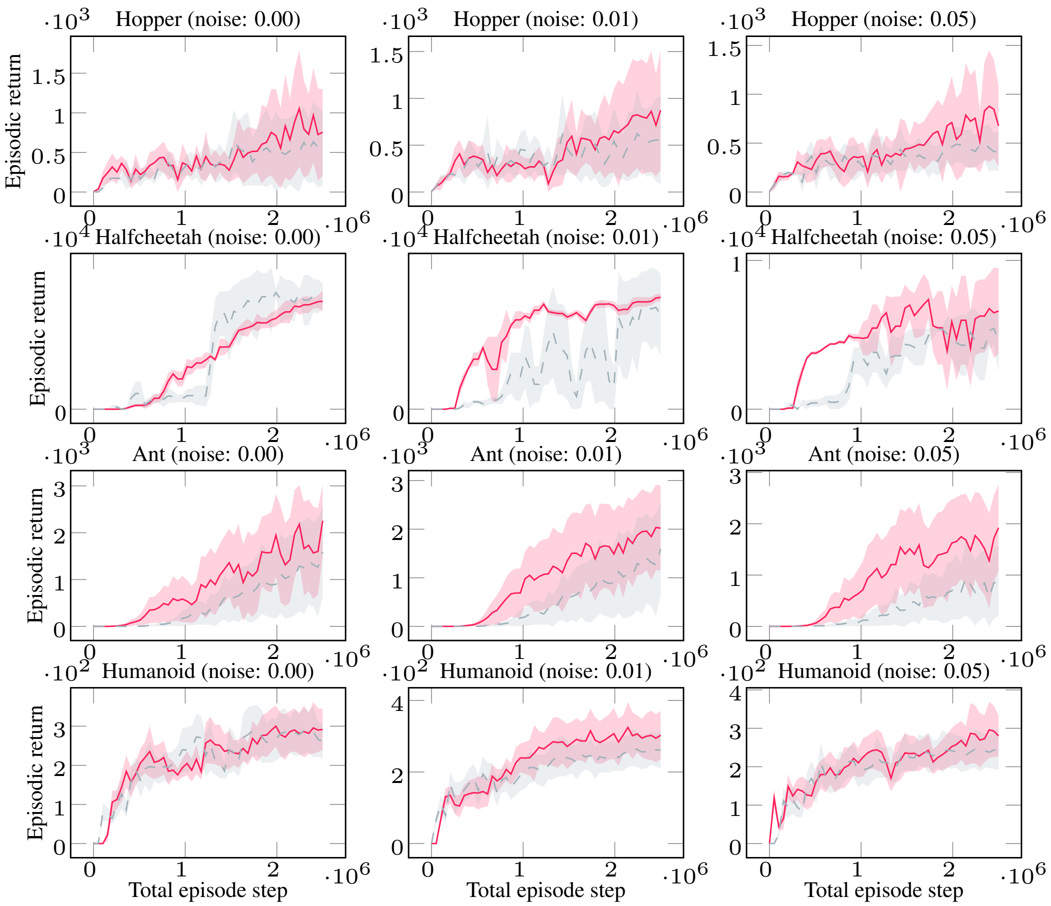


Figure 3: Performance of TD, and β-dTD on continuous control benchmark. These results were obtained using the A2C algorithm. Each column corresponds to different noise levels (coef $=$ $0.00, 0.01, 0.05$), and each row corresponds to different environments. The tuned β values for β-dTD were 0.24, 0.60, 075, 0.47 in Hopper, HalfCheetah, Ant, and Humanoid, respectively.

B.3 Details of the A2C Implementation

B.3.1 Learning Curves

B.3.2 Hyperparameters for A2C

The search space of the hyperparameters is summarized in Table 4. The values chosen finally are summarized in Tables **??**.

B.4 Computing Infrastructure and Reproducibility

Computing infrastructure Experiments were conducted on a machine with four NVIDIA Tesla V100 GPUs (32GB each) and an Intel Xeon E5-2698 v4 CPU. Although all experiments can be executed on a single GPU, multiple GPUs were used to run independent trials in parallel for efficiency.

Training time Hyperparameter tuning typically took 6–9 hours depending on the environment. Training time for the final runs depended on the environment and ranged from 10 to 60 minutes.

Reproducibility All experiments were conducted with the random seed fixed in the training scripts. However, MuJoCo (accessed via Brax) uses its own internal random seed that is not directly controllable, so full determinism cannot be ensured. The code is available at `https://github.com/4thhia/differential_TD` for reproducibility.

Table 4: Hyperparameter search space

Hyperparameter	Search Space
environment steps per update (number of parallel environment: 64)	$\{8, 16, 32\}$
number of epochs per update	range(5, 20)
minibatch size	$\{256, 512\}$
learning rate	$\log(\text{interval}(1e-6, 5e-3))$
normalize advantage	$\{\text{True}, \text{False}\}$
gae lambda	interval(0.8, 0.9999)
entropy coefficient	interval(0.0, 0.3)
value loss weight	interval(0.0, 1.0)
mixture raio β	interval(0.0, 1.0)

Table 5: Best hyperparameters for A2C with TD and β-dTD across environments

Hyperparameter	TD				β-dTD			
	Hopper	Halfcheetah	Ant	Humanoid	Hopper	Halfcheetah	Ant	Humanoid
environment steps/update	32	16	8	16	16	8	8	16
epochs/update	7	5	11	15	16	9	12	19
minibatch size	512	256	512	512	512	256	256	512
learning rate	1.18e-3	5.93e-4	3.71e-4	1.40e-3	1.88e-6	3.52e-4	2.90e-6	6.83e-7
normalize advantage	False	False	False	False	False	False	False	True
GAE lambda	0.886	0.833	0.935	0.999	0.890	0.950	0.960	0.827
entropy coefficient	0.121	0.018	0.162	0.021	0.031	5.38e-6	0.094	0.046
value loss weight	0.049	0.513	0.711	0.091	0.825	0.602	0.457	0.546
mixture ratio β	—	—	—	—	0.244	0.598	0.750	0.467

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We not only theoretically developed a model-free, TD method based on the HJB equation for continuous systems, but also validated the potential effectiveness empirically.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: On the theoretical aspect, we clearly noted that currently the proposed method is not applicable to off-policy settings. On the empirical side, we found that the proposed method might work similarly to the baseline method when the process noise is large.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The assumptions are clearly stated, and we added further explanations about them. Full proofs of the propositions are provided in the main text and in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All the experimental settings are described in the main text and the appendix. We also made the codes publicly available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We made the codes publicly available. The RL environment we used is originally publicly available.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the experimental settings are described in the main text and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The achieved rewards are compared considering the variance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the experimental settings are described in the main text and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The study is strongly from the theoretical perspective, and we do not think it is straightforward to discuss the direct societal impact at this stage.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We referred to the libraries we used in the experiment.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We gave description along with the codes.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: NA

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.