Policy Search via Bayesian Optimization with Temporal Difference Gaussian Processes

Armin Lederer[1] **Anuj Srivastava**[2] **Andreas Krause**[2]
[1] National University of Singapore [2] ETH Zurich
armin.lederer@nus.edu.sg
asrivastav@ethz.ch
krausea@ethz.ch

Abstract

Bayesian optimization (BO) is a method commonly used for policy search in problems with low-dimensional policy parameterizations. While it is generally considered data-efficient, existing BO approaches are agnostic to the sequential structure of the optimization objective induced by policy roll-outs. Thereby, valuable information is discarded that could improve the convergence of BO. We address this inefficiency by developing and rigorously analyzing a novel approach for BO that relies on a temporal difference learning formulation for discounted infinite-horizon value functions based on Gaussian process (GP) regression. We derive learning error bounds for the proposed temporal difference GPs, such that we can exploit upper confidence bounds to analyze the cumulative regret of our BO approach. This analysis is further refined by bounding the maximal information gain for our temporal difference GP model. In a comparison with a BO approach agnostic to the sequential structure and a reinforcement learning baseline on classic control benchmarks, we demonstrate the practical advantages of our method.

1 Introduction

Policy search is a class of reinforcement learning (RL) methods which is particularly popular in robotics [1; 2]. Policy search relies on policy parameterizations, often comparatively low-dimensional and tailored to specific tasks, leading to success in a large variety of real-world robotics problems, such as balancing tasks [3], stroke movements [4], object manipulation [5], and locomotion [6]. Among the different policy search techniques, Bayesian optimization (BO) [7; 8] has gained increasing attention due its strong theoretical foundations [9; 10] as well as the straightforward consideration of safety constraints [11; 12] and robustness requirements [13].

BO, also referred to as Gaussian process (GP) bandit optimization, is a class of black-box optimization algorithms that aims to find the global optimum given noisy samples of the objective function. For this purpose, it sequentially learns a model of the objective using GP regression [14], based on which an acquisition function proxy is constructed for optimization. Using suitable acquisition functions, this approach can be shown to achieve order optimal regret [15; 16] and sample complexities [17]. Since we can interpret policy search problems via the lens of black-box optimization, BO algorithms can be immediately applied to them. In particular, BO has been demonstrated to be highly effective in applications with low-dimensional, structured policy parameterizations, such as PID [18] and LQR parameter optimization [19], and the automatic tuning of model predictive control [20]. While there has been a considerable focus on suitable policy parameterizations [21; 19; 22; 23], existing BO approaches for policy search remain agnostic to the underlying sequential structure of returns induced by roll-outs. By ignoring such an informative structure, valuable insight is discarded that would beneficially affect the learning of value functions and thereby improve the convergence of BO.

2nd Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2025).

Contributions: We address this weakness of existing BO approaches by proposing and analyzing a novel upper confidence bound algorithm that relies on a Gaussian process model of the value function learned from its temporal differences. In particular, our key contributions are the following:

- **Temporal difference GPs:** We propose a novel approach for temporal difference learning of discounted infinite-horizon value functions with GPs by exploiting the closedness of GPs under linear operators. Under weak assumptions on the transition probabilities, we extend existing probabilistic regression error bounds to the obtained value function estimate.
- **Regret bounds:** Based on the derived error bound, we extend the common upper confidence bound (UCB) approach in BO [9] to maximizing discounted infinite-horizon value functions. By relating the GP variances of the value function to those of temporal differences, we bound the regret of the proposed UCB algorithm in terms of the maximum information gain.
- **Information gain analysis:** To establish regret bounds directly in terms of the number of episodes, we provide a novel analysis of the information gain given our GP model exploiting the temporal difference structure via a matrix representation. We show that our regret bounds are competitive to those of related model-based RL approaches.

The remainder of this paper is structured as follows: We formalize our problem setting in Section 2. Our approach for policy search via BO with temporal difference GP is proposed and analyzed in Section 3. In Section 4, we discuss the connection of our method to related work. We compare our method to different baselines on benchmark problems in Section 5, before we conclude the paper in Section 6.

2 Problem Statement

We consider a discrete-time stochastic dynamical system

$$\boldsymbol{s}_{t+1} \sim p(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t) \qquad \boldsymbol{s}_0 \sim \rho \tag{1}$$

where $\boldsymbol{s}_t \in \mathcal{S} \subset \mathbb{R}^{d_s}$ is the state, $\boldsymbol{a}_t \in \mathcal{A} \subset \mathbb{R}^{d_a}$ is the action, $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel and ρ is some initial state distribution. We assume that the transition kernel p is unknown, but we restrict its distribution to satisfy the following properties.

Assumption 1. *The transition kernel $p(\cdot|\boldsymbol{s}, \boldsymbol{a})$ is compactly supported such that $\mathcal{S}$ is a compact set. Moreover, the process noise $\boldsymbol{w} = \boldsymbol{s}^+ - \mathbb{E}_{\boldsymbol{s}^+}[\boldsymbol{s}^+]$ is sub-Gaussian with $\sup_{\boldsymbol{v}:\|\boldsymbol{v}\|=1} \mathbb{E}_{\boldsymbol{w}}[\exp((\boldsymbol{v}^T\boldsymbol{w})^2/\lambda^2)] \leq 2$ for $\lambda \in \mathbb{R}_{\geq 0}$.*

The assumption of a compact support of p ensures that trajectories generated by (1) cannot diverge indefinitely, such that learning can be realized on a bounded set. Thereby, this requirement resembles similar assumptions in literature on RL, e.g., [24]. The restriction on noise is commonly found in literature on Bayesian optimization [9; 10] and reinforcement learning approaches [25; 26]. It poses an upper bound on the decay of the tails of the distribution of $\boldsymbol{w}$. Since this assumption admits a wide range of distributions such as Gaussian and uniform distributions, it is generally not considered restrictive.

We study the problem of determining a policy $\boldsymbol{\pi} : \mathcal{S} \times \Theta \to \mathcal{A}$ parameterized by $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^{d_\theta}$ with compact set Θ which maximizes the discounted average cumulative returns

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}_0}\left[V(\boldsymbol{s}_0, \boldsymbol{\theta})\right] = \mathbb{E}_{\boldsymbol{s}_0}\left[\mathbb{E}_{\boldsymbol{s}_1, \boldsymbol{s}_2, \dots}\left[\sum\nolimits_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{\pi}(\boldsymbol{s}_t, \boldsymbol{\theta}))\middle| \boldsymbol{s}_0\right]\right], \tag{2}$$

where $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is an immediate reward, $\gamma \in (0, 1)$ is a discount factor, and the sequence of states $\boldsymbol{s}_t$ is generated by (1) with $\boldsymbol{a}_t = \boldsymbol{\pi}(\boldsymbol{s}_t, \boldsymbol{\theta})$. Since p is unknown, we aim to iteratively learn the optimal policy parameter $\boldsymbol{\theta}^*$ which maximizes (2) by interacting with the dynamics (1). In particular, we consider an episodic setting in which we roll out a policy $\boldsymbol{\pi}$ with parameter $\boldsymbol{\theta}^n$ for $M_n \in \mathbb{N}$ time steps in every episode $n = 1, \dots, N$. This results in a trajectory $\tau^n = \{(\boldsymbol{s}_0^n, r_0^n), (\boldsymbol{s}_1^n, r_1^n), \dots, (\boldsymbol{s}_{M_n}^n, r_{M_n}^n)\}$ with $r_i^n = r(\boldsymbol{s}_i^n, \boldsymbol{\pi}(\boldsymbol{s}_i^n, \boldsymbol{\theta}^n))$ for each episode n, which we aggregate into a data set $\mathbb{D}^N = \{(\tau^n)_{n=1,\dots,N}\}$. This data sets is subsequently used to learn an estimate of the returns $J(\cdot)$ defined in (2) based on GP regression.

The goal of this paper is the development of an algorithm for determining the parameter $\boldsymbol{\theta}^{n+1}$ for the next episode given the data $\mathbb{D}^n$ obtained so far. We measure the performance of this algorithm via the commonly used metric of cumulative regret [27]

$$R_N = \sum\nolimits_{n=1}^{N} J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_n), \tag{3}$$

where $\boldsymbol{\theta}^*$ denotes the maximizer of (2). Therefore, an intuitive requirement for a useful algorithm choosing $\boldsymbol{\theta}^n$ is $R_N/N \to 0$ for $N \to \infty$, i.e., the cumulative regret must be sub-linear.

To formally analyze the asymptotic behavior of the cumulative regret, additional assumptions about the problem complexity are necessary. Since we employ Gaussian process regression for learning an estimate of the returns $J(\cdot)$, we pose this requirement in terms of the function complexity as measured through the kernel used for regression.

Assumption 2. *Let $k : (\mathcal{S} \times \Theta) \times (\mathcal{S} \times \Theta) \to \mathbb{R}_{\geq 0}$ be a kernel with bounded Hessian* $\mathrm{trace}\big(\nabla_{\boldsymbol{s}} \nabla_{\boldsymbol{s}'}^T k([\boldsymbol{s};\boldsymbol{\theta}], [\boldsymbol{s}';\boldsymbol{\theta}])\big|_{\boldsymbol{s}'=\boldsymbol{s}}\big) \leq L_k$ *for all $\boldsymbol{s}, \boldsymbol{\theta} \in \mathcal{S} \times \Theta$. The value function $V : \mathcal{S} \times \Theta \to \mathbb{R}$ has a bounded norm in the reproducing kernel Hilbert space (RKHS) defined through the kernel k, i.e., $\|V\|_k = \sqrt{\langle V, V \rangle} \leq B$ for some $B \in \mathbb{R}_{>0}$.*

The restriction to functions with bounded RKHS norm can be commonly found in research on Bayesian optimization [9; 10] and reinforcement learning [25; 26] as it allows the analysis of learning errors. Depending on the used kernel, the RKHS can contain large function classes, e.g., analytic and Sobolev functions [28; 29]. Due to known continuity properties of value functions, e.g., [30; 31], this restriction is generally not severely restricting the applicability of our results.

3 Bayesian Optimization of Discounted Infinite-Horizon Value Functions

To iteratively select the parameters $\boldsymbol{\theta}$ in each episode, we develop and analyze a novel variant of the GP-UCB algorithm that exploits structured Gaussian processes inspired by temporal difference learning. For this purpose, we firstly introduce some general background on Gaussian process regression in Section 3.1. In Section 3.2, we propose a structured GP prior that enables learning value functions $V(\cdot,\cdot)$ from trajectory data taking rewards as training targets. Based on the resulting temporal difference GP models, we present a Bayesian optimization algorithm based on upper confidence bounds in Section 3.3 and derive cumulative regret bounds in Section 3.4. To render these bounds fully data-dependent, we analyze the maximum information gain under our structured GP prior in Section 3.5.

3.1 Gaussian Process Regression

Gaussian process regression [14] is a supervised machine learning technique that relies on Bayesian principles. A Gaussian process $g \sim \mathcal{GP}(m_g, k_g)$ can be considered a generalization of the Gaussian distribution to functions $g : \mathbb{R}^{d_x} \to \mathbb{R}$, which is fully specified through the prior mean $\mathbb{E}[g(\boldsymbol{x})] = m_g(\boldsymbol{x})$, $m_g : \mathbb{R}^{d_x} \to \mathbb{R}$, and a covariance function $\mathrm{Cov}(g(\boldsymbol{x}), g(\boldsymbol{x}')) = k_g(\boldsymbol{x}, \boldsymbol{x}')$ defined through a kernel $k_g : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \to \mathbb{R}_{\geq 0}$. Given a data set $\mathbb{D} = \{(\boldsymbol{x}_n, y_n)_{n=1,\ldots,N}\}$ with training targets $y_n = g(\boldsymbol{x}_n) + \epsilon_n$ perturbed by i.i.d. Gaussian noise $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$, we can formulate the joint distribution between a test output $g(\boldsymbol{x})$ and the training targets $\boldsymbol{y}$ as

$$\begin{bmatrix} \boldsymbol{y}_N \\ g(\boldsymbol{x}) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{m}_g \\ m_g(\boldsymbol{x}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_g(\boldsymbol{X}_N) + \sigma^2 \boldsymbol{I} & \boldsymbol{k}_g(\boldsymbol{x}) \\ \boldsymbol{k}_g^T(\boldsymbol{x}) & k_g(\boldsymbol{x}, \boldsymbol{x}) \end{bmatrix} \right), \tag{4}$$

where $[\boldsymbol{y}_N]_i = y_i$, $[\boldsymbol{X}_N]_i = \boldsymbol{x}_i$, $[\boldsymbol{m}_g]_i = m_g(\boldsymbol{x}_i)$, $[\boldsymbol{K}_g(\boldsymbol{X}_N)]_{i,j} = k_g(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $[\boldsymbol{k}_g(\boldsymbol{x})]_i = k_g(\boldsymbol{x}_i, \boldsymbol{x})$ for $i, j = 1, \ldots, N$. Due to this joint distribution, the posterior conditioned on the data set is a Gaussian $g(\boldsymbol{x})|\mathbb{D} \sim \mathcal{N}(\mu_g(\boldsymbol{x}), \sigma_g^2(\boldsymbol{x}))$ with mean and variance

$$\mu_g(\boldsymbol{x}) = m_g(\boldsymbol{x}) + \boldsymbol{k}_g^T(\boldsymbol{x})(\boldsymbol{K}_g(\boldsymbol{X}_N) + \sigma^2 \boldsymbol{I})^{-1}(\boldsymbol{y}_N - \boldsymbol{m}_g), \tag{5}$$

$$\sigma_g^2(\boldsymbol{x}) = k_g(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_g^T(\boldsymbol{x})(\boldsymbol{K}_g(\boldsymbol{X}_N) + \sigma^2 \boldsymbol{I})^{-1}\boldsymbol{k}_g(\boldsymbol{x}). \tag{6}$$

The matrix $\boldsymbol{K}_g(\boldsymbol{X}_N) + \sigma^2 \boldsymbol{I}$ is not only crucial for GP regression, but also has an information-theoretic meaning as it reflects the information gain due to data. Thus, the maximum information gain

$$\Gamma_{k_g}(N) = \max_{\boldsymbol{x}_n \in \mathcal{X}, n=1,\ldots,N} \frac{1}{2} \log\det(\boldsymbol{I} + \sigma^{-2} \boldsymbol{K}_g(\boldsymbol{X}_N)) \tag{7}$$

acts as a measure for the complexity of learning problems [9].

3.2 Learning Value Functions using Temporal Difference Gaussian Processes

Since we do not have access to measurements of $V(\boldsymbol{s},\boldsymbol{\theta})$, we cannot directly apply GP regression to learn a model of the value function. Therefore, we take inspiration from model-free reinforcement learning, where value functions are commonly learned from temporal differences [27]

$$\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\boldsymbol{\theta}) = V(\boldsymbol{s},\boldsymbol{\theta}) - \gamma V(\boldsymbol{s}^+,\boldsymbol{\theta}). \tag{8}$$

For notational simplicity, we use here the shorthand notation $\boldsymbol{s}^+ \sim p(\boldsymbol{s}^+|\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$. By fitting a model for the temporal difference (8), $r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$ becomes available as unbiased training target for regression because $\mathbb{E}_{\boldsymbol{s}^+}[\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\theta)] = r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$. Moreover, we can exploit the closedness of GPs under the linear operation (8) [32; 33], which allows us to infer the conditional distribution of V given noisy measurements of ΔV by formulating their joint distribution analogously to (4).

For this purpose, we define a structured GP prior for ΔV by assuming $V \sim \mathcal{GP}(m_V, k_V)$. Exploiting linearity of the expectation, we immediately obtain $\mathbb{E}[\Delta V(\boldsymbol{z})] = m_V(\boldsymbol{x}) - \gamma m_V(\boldsymbol{x}^+)$, where $\boldsymbol{x} = [\boldsymbol{s};\boldsymbol{\theta}]$, $\boldsymbol{x}^+ = [\boldsymbol{s}^+;\boldsymbol{\theta}]$, $\boldsymbol{z} = [\boldsymbol{x};\boldsymbol{x}^+]$, and we define $V(\boldsymbol{x}) = V(\boldsymbol{s},\boldsymbol{\theta})$ and $\Delta V(\boldsymbol{z}) = \Delta V(\boldsymbol{x},\boldsymbol{x}^+,\boldsymbol{\theta})$ with slight abuse of notation. Similarly, the definition of ΔV immediately yields

$$\mathrm{Cov}(\Delta V(\boldsymbol{z}_i),\Delta V(\boldsymbol{z}_j)) = k_V(\boldsymbol{x}_i,\boldsymbol{x}_j) - \gamma k_V(\boldsymbol{x}_i,\boldsymbol{x}_j^+) - \gamma k_V(\boldsymbol{x}_i^+,\boldsymbol{x}_j) + \gamma^2 k_V(\boldsymbol{x}_i^+,\boldsymbol{x}_j^+) \tag{9}$$

for $V \sim \mathcal{GP}(m_V, k_V)$. Thus, we obtain the structured prior $\Delta V \sim \mathcal{GP}(m_\Delta, k_\Delta)$ with $m_\Delta(\boldsymbol{z}) = \mathbb{E}[\Delta V(\boldsymbol{z})]$ and $k_\Delta(\boldsymbol{z}_i,\boldsymbol{z}_j) = \mathrm{Cov}(\Delta V(\boldsymbol{z}_i),\Delta V(\boldsymbol{z}_j))$. Finally, we have $\mathrm{Cov}(V(\boldsymbol{x}_i),\Delta V(\boldsymbol{z}_j)) = k_V(\boldsymbol{x}_i,\boldsymbol{x}_j) - \gamma k_V(\boldsymbol{x}_i,\boldsymbol{x}_j^+)$, such that the joint distribution between measurements with Gaussian noise, i.e., $\epsilon = \Delta V(\boldsymbol{s},\boldsymbol{s}^+,\theta) - r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta})) \sim \mathcal{N}(0,\sigma^2)$, and the value function is given by

$$\begin{bmatrix}\boldsymbol{r}\\ V(\boldsymbol{x})\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{m}_\Delta\\ m_V(\boldsymbol{x})\end{bmatrix}, \begin{bmatrix}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)+\sigma^2\boldsymbol{I} & \boldsymbol{k}_V(\boldsymbol{x}) - \gamma\boldsymbol{k}_V^+(\boldsymbol{x})\\ (\boldsymbol{k}_V(\boldsymbol{x}) - \gamma\boldsymbol{k}_V^+(\boldsymbol{x}))^T & k_V(\boldsymbol{x},\boldsymbol{x})\end{bmatrix}\right) \tag{10}$$

where $[\boldsymbol{k}_V(\boldsymbol{x})]_i = k_V(\boldsymbol{x}_i,\boldsymbol{x})$, $[\boldsymbol{k}_V^+(\boldsymbol{x})]_i = k_V(\boldsymbol{x}_i^+,\boldsymbol{x})$, $[\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)]_{i,j} = k_\Delta(\boldsymbol{z}_i,\boldsymbol{z}_j)$, and $[\boldsymbol{r}]_i = r(\boldsymbol{s}_i,\boldsymbol{\pi}(\boldsymbol{s}_i,\boldsymbol{\theta}_i))$. Note that for a data set $\mathbb{D}^N$ consisting of N trajectories τ^n with length M_n, we have $\boldsymbol{x}_i = \boldsymbol{x}_{m_i}^{n_i}$ and $\boldsymbol{x}_i^+ = \boldsymbol{x}_{m_i+1}^{n_i}$ for $n_i = \min_{\sum_{k=1}^{n'} M_k \geq i} n'$ and $m_i = i - 1 - \sum_{k=1}^{n_i-1} M_k$ for $i = 1,\ldots,\sum_{n=1}^{N} M_n$. Based on the joint distribution (10), we can proceed analogously to Section 3.1 to obtain the posterior $V(\boldsymbol{x})|\mathbb{D}^N \sim \mathcal{N}(\mu_V(\boldsymbol{x}),\sigma_V^2(\boldsymbol{x}))$ with

$$\mu_V(\boldsymbol{x}) = m_V(\boldsymbol{x}) + (\boldsymbol{k}_V(\boldsymbol{x}) - \gamma\boldsymbol{k}_V^+(\boldsymbol{x}))^T(\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)+\sigma^2\boldsymbol{I})^{-1}(\boldsymbol{r} - \boldsymbol{m}_\Delta), \tag{11}$$

$$\sigma_V^2(\boldsymbol{x}) = k_V(\boldsymbol{x},\boldsymbol{x}) - (\boldsymbol{k}_V(\boldsymbol{x}) - \gamma\boldsymbol{k}_V^+(\boldsymbol{x}))^T(\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)+\sigma^2\boldsymbol{I})^{-1}(\boldsymbol{k}_V(\boldsymbol{x}) - \gamma\boldsymbol{k}_V^+(\boldsymbol{x})). \tag{12}$$

Note that we assume Gaussian noise ϵ purely for the derivation of posterior GP expressions (11) and (12), but this assumption is generally not satisfied when the randomness stems from transition dynamics (1) due to nonlinearity of V. Therefore, existing theoretical results [34] do not apply directly to (11) and (12). We overcome this limitation by showing that ϵ is sub-Gaussian for sufficiently smooth value functions V and sub-Gaussian process noise $\boldsymbol{w} = \boldsymbol{s}^+ - \mathbb{E}_{\boldsymbol{s}^+}[\boldsymbol{s}^+]$, such that existing frequentist error bounds for GP regression become applicable [9; 10; 35].[1]

Proposition 1. *Consider a stochastic dynamical system* (1) *satisfying Assumption 1, assume that the value function V satisfies Assumption 2, and let $m_V \equiv 0$.*[2] *Then, the error for learning V from a data set $\mathbb{D}^n$ using Gaussian process regression with posterior mean* (11) *and variance* (12) *is bounded by*

$$|\mu_V(\boldsymbol{x}) - V(\boldsymbol{x})| \leq \beta(\delta)\sigma_V(\boldsymbol{x}) \tag{13}$$

jointly for all $\boldsymbol{x},\boldsymbol{\theta}$ in a compact domain $\mathcal{S}\times\Theta$ with probability $1-\delta$, $\delta \in (0,1)$, *where*

$$\beta(\delta) = 2\sqrt{d_s L_k}B\lambda\sqrt{\log\det\left(\boldsymbol{I} + \frac{1}{\sigma^2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)\right) - 2\log(\delta) + \sigma B}. \tag{14}$$

3.3 Policy Search via Bayesian Optimization

Based on this GP approach for learning value functions, we interpret reinforcement learning as a black-box optimization problem with unknown objective $J(\cdot)$ defined in (2). Due to the regression

[1] Proofs for all theoretical results can be found in the appendix.

[2] The extension to non-zero prior mean only requires $V(\cdot) - m_V(\cdot)$ to exhibit bounded RKHS norm.

Algorithm 1 Temporal Difference Gaussian Process Upper Confidence Bound Optimization

```
$\hat{J}_1(\boldsymbol{\theta}) \leftarrow \mathbb{E}_{\boldsymbol{s}_0}[B\sqrt{k_V([\boldsymbol{x}, \boldsymbol{x}])}]$ with $\boldsymbol{x} = [\boldsymbol{s}_0^T, \boldsymbol{\theta}^T]^T$
Initialize $\mathbb{D}^1$ (e.g., $\mathbb{D}^1 \leftarrow \emptyset$)
for $n = 1, \dots, N$ do
    Get optimistic policy parameters $\boldsymbol{\theta}_n \leftarrow \arg\max_{\boldsymbol{\theta} \in \Theta} \hat{J}_n(\boldsymbol{\theta})$
    Roll-out $\boldsymbol{\pi}(\cdot, \boldsymbol{\theta}_n)$ for $M_n$ steps and record $\tau^n = \{(\boldsymbol{s}_0^n, r_0^n), \dots, (\boldsymbol{s}_{M_n}^n, r_{M_n}^n)\}$
    Extend data set $\mathbb{D}^{n+1} \leftarrow \mathbb{D}^n \cup \tau^n$
    Determine $\mu_{V,n+1}$, $\sigma_{V,n+1}^2$, $\beta(\frac{6\delta}{\pi^2(n+1)^2})$ for data set $\mathbb{D}^{n+1}$ using (11), (12), (30)
    Define optimistic value function $\hat{V}_{n+1}(\boldsymbol{s}, \boldsymbol{\theta}) \leftarrow \mu_{V,n+1}(\boldsymbol{x}) + \beta(\frac{6\delta}{\pi^2(n+1)^2})\sigma_{V,n+1}(\boldsymbol{x})$
    Compute $\hat{J}_{n+1}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}_0}[\hat{V}_{n+1}(\boldsymbol{s}_0, \boldsymbol{\theta})]$
end for
```

error bound for temporal difference learning derived in Proposition 1, this interpretation immediately allows the extension of upper confidence bound algorithms [9; 36] to our setting. To highlight this connection, we refer to the resulting algorithm, which is outlined in Algorithm 1, as **T**emporal **D**ifference **G**aussian **P**rocess **U**pper **C**onfidence **B**ound Optimization (**TDGP-UCB**).

Starting from the prior GP model of the value function, our approach computes the expectation of the upper confidence bound of our value function estimate over the initial state distribution

$$\hat{J}_n(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}_0}\left[\mu_{V,n}\left([\boldsymbol{s}_0; \boldsymbol{\theta}]\right) + \beta(\tfrac{6\delta}{\pi^2 n^2})\sigma_{V,n}\left([\boldsymbol{s}_0; \boldsymbol{\theta}]\right)\right]. \tag{15}$$

Note that we select $\delta_n = {}^{6\delta}/_{\pi^2 n^2}$ in the n-th episode such that the union bound guarantees the upper confidence bound to hold jointly for all $n \in \mathbb{N}$ with probability $1 - \delta$. Finally, we maximize the optimistic estimate $\hat{J}_n(\cdot)$ with respect to the policy parameters

$$\boldsymbol{\theta}_n = \arg\max_{\boldsymbol{\theta} \in \Theta} \hat{J}_n(\boldsymbol{\theta}) \tag{16}$$

and update our GP model using the trajectory data obtained from the roll-out, such that the next episode can start with an improved model. While the optimization problem (16) is generally non-convex, we assume to have access to an oracle which provides us with the global maximum as common in the Bayesian optimization literature [37].

3.4 Regret Bounds

While Algorithm 1 barely differs from standard upper confidence bound approaches in Bayesian optimization [9; 10], the theoretical analysis in these works does not immediately extend to it. The reason for this difficulty lies in the fact that the regression error bound (29) relies on the posterior variances $\sigma_V^2(\boldsymbol{x})$. The sum of these variances cannot directly be bounded through the maximum information gain analogously as in standard Bayesian optimization [9] since it relies on different covariance functions ($k_V(\cdot, \cdot)$ and $k_{\Delta V}(\cdot, \cdot)$). We overcome this challenge by bounding the variance $\sigma_V^2(\boldsymbol{x})$ through the sum of GP variances

$$\sigma_\Delta^2(\boldsymbol{z}) = k_\Delta(\boldsymbol{z}, \boldsymbol{z}) - \boldsymbol{k}_\Delta^T(\boldsymbol{z})(\boldsymbol{K}_\Delta(\boldsymbol{Z}_N) + \sigma^2 \boldsymbol{I})^{-1}\boldsymbol{k}_\Delta(\boldsymbol{z}) \tag{17}$$

along the roll-out trajectories and a residual error. Due to the discount factor, we can make the residual error arbitrarily small using trajectories with sufficiently many time steps M_n in every episode n. By increasing this roll-out length M_n with growing n, this enables us to obtain the following result.

Theorem 1. *Consider a dynamical system* (1) *satisfying Assumption 1, such that Assumption 2 holds for the value function $V(\cdot)$. Assume that parameters $\boldsymbol{\theta}_{\mathrm{UCB}}^n$ are chosen via* (16) *with GP mean* (11) *and variance* (12) *learned from a data set $\mathbb{D}^n$ which is obtained from roll-outs of length M_n, such that $\gamma^{M_n} \leq {}^{\kappa}/_{n^2}$ for $\kappa \in (0, 1)$ in each episode $n = 1, \dots, N$. Then, the regret* (3) *satisfies*

$$R_N \leq c\sqrt{NM_N \Gamma_{k_\Delta}\left(NM_N\right)\left(\Gamma_{k_\Delta}\left(NM_N\right) + \log(N)\right)} \tag{18}$$

with probability $1 - \delta$ and constant $c \in \mathbb{R}_{\geq 0}$ for $N > 1$.

While we do not explicitly state it, the constant c depends on the choice of the parameter κ. However, this dependency does not have any asymptotic effect since c behaves as $\mathcal{O}(1)$ for $\kappa \to 0$ due to other

Table 1: Asymptotic regret bounds for common base kernels k_V with $M_N \in \mathcal{O}(\log(N))$.

linear kernel	SE kernel	Matérn kernel with param. ν
$\mathcal{O}^*\left(dN^{1/2}\right)$	$\mathcal{O}^*\left(N^{1/2}\log(N)^d\right)$	$\mathcal{O}^*\left((N\log(N))^{\frac{1}{2}\frac{(3d(d+1)+2\nu)}{(2\nu+d(d+1))}}\right)$

remaining constant dependencies. For $\kappa \to 1$, the constant c behaves as $\mathcal{O}(\kappa)$, i.e., it again converges to a constant. Note that κ neither has an asymptotic effect on M_N: the satisfaction of $\gamma^{M_N} \leq {}^{\kappa}/_{N^2}$ requires $M_N \geq \frac{\log(\kappa)}{\log(\gamma)} - \frac{2\log(N)}{\log(\gamma)}$, i.e., κ essentially defines an offset for the number of steps during the roll-outs. In contrast, the necessary growth rate is independently given by $\log(N)$. If we limit the growth rate of M_N to this necessary one, we can simplify the bound in (18) as shown in the following.

Corollary 1. *Assume that the assumptions of Theorem 1 are satisfied and $M_N \in \mathcal{O}(\log(N))$ holds. Then, with probability $1-\delta$, the regret* (3) *satisfies*

$$R_N \in \mathcal{O}^*\left(\sqrt{N\Gamma^2_{k_\Delta}(N\log(N))}\right) \tag{19}$$

where $\mathcal{O}^*$ denotes asymptotic expressions up to dimension-independent logarithmic factors.

3.5 Maximum Information Gain Bounds

To finally quantify the asymptotic behavior of the regret for Algorithm 1, it remains to bound the maximum information gain $\Gamma_{k_\Delta}(\cdot)$. Even though bounds for many commonly used kernels [9; 38; 39] and certain combinations of them [40] exist, the special structure of the temporal difference kernel $k_\Delta(\cdot,\cdot)$ defined using (9) prevents the direct application of these results. Nevertheless, the temporal difference kernel $k_\Delta(\cdot,\cdot)$ is still a composition of base kernels $k_V(\cdot,\cdot)$, which is also reflected by the Gram matrix $\boldsymbol{K}_\Delta(\boldsymbol{Z}_N))$, i.e., the core element of the definition of the maximum information gain in (7). Taking inspiration from early work on model-free RL with GPs [41], we see that the Gram matrix $\boldsymbol{K}_\Delta(\boldsymbol{Z}_N))$ can indeed be expressed through the Gram matrix $K_V(\tilde{\boldsymbol{X}}_N)$ of the base kernel $k(\cdot,\cdot)$ via $\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)) = \Xi K_V(\tilde{\boldsymbol{X}}_N)\Xi^T$. Here, the matrix $\Xi = \mathrm{blkdiag}(\Xi_1,\dots,\Xi_N)$ with blocks

$$\Xi_n = \begin{bmatrix} 1 & -\gamma & 0 & \dots & 0 \\ 0 & 1 & -\gamma & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -\gamma \end{bmatrix} \tag{20}$$

encodes the temporal difference structure of the data with each block corresponding to one roll-out trajectory. By exploiting bounds for the eigenvalues of tri-diagonal Toeplitz matrices [42], we can show that these the norm of these matrices satisfies $\|\Xi\| \leq (1+\gamma)$. This property essentially allows us to lump the effect of the temporal difference structure into the noise parameter in (7), such that bounds for the maximum information gain $\Gamma_{k_\Delta}(\cdot)$ asymptotically behave the same as $\Gamma_{k_V}(\cdot)$. We formalize this insight in the following result.

Theorem 2. *The maximum information gain for the temporal difference kernel k_Δ is asymptotically bounded by the information gain of its base kernel, i.e.,*

$$\Gamma_{k_\Delta}(NM_N) \in \mathcal{O}(\Gamma_{k_V}(NM_N)) \tag{21}$$

for a non-decreasing sequence M_n, $n = 1,\dots,N$.

By combining this result with Corollary 1, we immediately obtain the regret bound $\mathcal{O}^*(N^{1/2}\Gamma_{k_V}(N\log(N)))$. This allows us to formulate explicit regret bounds for many frequently used base kernels $k_V(\cdot,\cdot)$ as illustrated in Table 1. Moreover, it enables a straightforward comparison with related approaches in the literature. For example, it can be immediately observed that our asymptotic regret bound is slightly worse compared to the original GP-UCB algorithm, which yields $\mathcal{O}^*(N^{1/2}\Gamma_{k_V}(N))$ [9]. The extra $\log(N)$ term in our regret bound is likely to be an artifact of the delayed update of the GP model after a roll-out has generated multiple data points. This artifact can be similarly observed in model based RL formulations [25], for which $\mathcal{O}^*(N^{1/2}\Gamma_{k_V}(N\log(N)))$ can also be shown when ignoring dimension-dependent factors. Thus, the proposed TDGP-UCB algorithm ensures the same asymptotic regret bound as state-of-the-art model-based RL methods relying on GP models, which underlines its efficiency.

4 Related Work

Numerous variants and special cases of Bayesian optimization have been proposed for policy search. Many works focus on the parameterization of policies, among which LQR policies are probably the most widely investigated. Developed approaches range from the optimization over controller parameters [21], LQR weight matrices [19], dynamics parameters for LQR [22], and combinations thereof [23]. In addition to the parameterization, approaches have been tailored to specific control architectures [43], and properties of GPs have been exploited to combine simulation and real-world data, such that the data efficiency of policy search can be improved [44; 45; 46]. However, it has only been noticed recently that the sequential structure of policy search problems can be incorporated into the prior GP mean function to reduce the sample complexity [47].

The general idea of combining temporal difference learning with GP models to infer discounted infinite-horizon value functions originates from the field of Bayesian RL [48]. It has been employed in a variety of GP-based RL algorithms including SARSA [49] and Q-learning [50; 51]. While the GP model considered in early work [41] seems to exhibit a high similarity with our TDGPs, it puts the GP prior on the random samples of value functions instead of the value functions themselves. This conceptually different idea does not allow the derivation of regression error bounds. Consequently, no theoretical results comparable to ours have been derived for the discussed Bayesian RL techniques.

In contrast to Bayesian RL, upper confidence bounds play a crucial role in many theoretical RL approaches that employ function approximation. These methods often rely on a form of optimistic least squares value iteration, i.e., the value function is not directly obtained via a form of TD learning. These approaches allow the derivation of strong theoretical guarantees similarly as for our method [52; 53]. For example, in episodic problems with kernelized linear MDPs, which are strongly related to our problem setting due to the equivalence of linear MDPs and linear value function models for finite state spaces [54], regret bounds have been derived for multiple such algorithms, e.g., variants of Q-learning [55; 56; 57]. When the transition probability is in the RKHS of the used kernel, uncertainty sampling can be employed to find ϵ-optimal policies for discounted infinite-horizon problems [58]. However, these approaches are purely theoretical and their practical effectiveness is unclear.

Note that the practical effectiveness of learning the value function in dependency of the policy parameters has been demonstrated experimentally. State- and parameter-dependent value functions are learned using deep RL in [59]. Moreover, policy evaluation networks only depending on a 'fingerprint' of a policy neural network are successfully employed in [60]. These examples further support our rationale of learning policy parameter-dependent value functions.

5 Numerical Evaluation

We demonstrate the effectiveness of the proposed TDGP-UCB algorithm in modified benchmark problems from the Gymnasium Classic Control Suite [61]. In Section 5.1, we describe the setting that is used for evaluation and we outline the baselines that we compare against. In Section 5.2, we present the results for experiments with dense rewards. Finally, we illustrate the effectiveness of our approach in experiments with sparse rewards in Section 5.3.

5.1 Simulation Setting and Baselines

As linear policies have been demonstrated to achieve a competitive performance on many RL benchmarks [62], we also adopt this policy class by choosing $\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{s}$, with bounded parameters $\boldsymbol{\theta} \in [-1,1]^{d_s}$. We apply these policies for $100 + \log(n)$ time steps in episode n of every environment and determine returns using $\gamma = 0.9$. We evaluate the performance of all algorithms using the cumulative regret averaged over 5 runs with different random seeds. To compute the regret, we find the optimal parameters $\boldsymbol{\theta}^*$ by numerically maximizing the returns using the Nelder-Mead method.

Since each episode generates more than 100 training samples for the temporal difference GP in Algorithm 1, data quickly accumulates. Therefore, we an approximation of the squared exponential kernel based on 200 random spectral features [63] as our base kernel $k_V(\cdot,\cdot)$. If not stated differently, we start each run with the roll-out of a random parameter $\boldsymbol{\theta}$. The hyperparameters of the approximated squared exponential kernel are tuned using log-likelihood maximization after 10 and 20 subsequent episodes, respectively. As common in the Bayesian optimization literature [64], we fix $\beta(\delta) = 2$. Moreover,

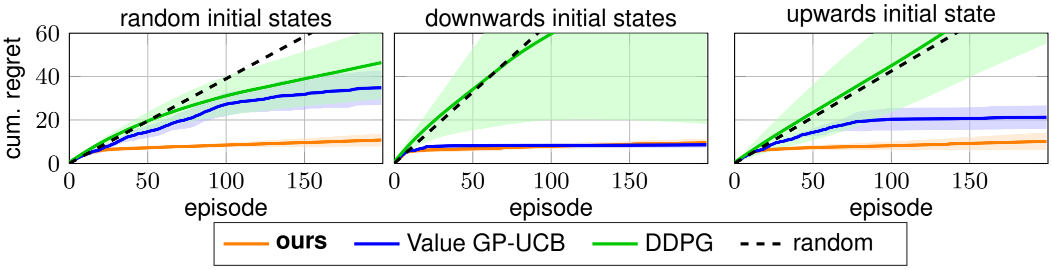

Figure 1: Average cumulative regret for the Gymnasium Pendulum environment with dense rewards. Shaded areas illustrate one standard deviation confidence intervals.

we approximate the expectation in (15) via the empirical mean over 10 samples that are fixed at the beginning of each run. The maximization in (16) is executed using the Nelder-Mead method.

We compare our TDGP-UCB algorithm to two baselines:

- **Value GP-UCB:** To demonstrate the benefits of exploiting the temporal difference decomposition of value functions in the GP model, we compare to a variant of GP-UCB that directly uses the value function estimates as training targets. For this, we perform Gaussian process regression on $V \sim \mathcal{GP}(m_V, k_V)$ using the dataset $\tilde{\mathbb{D}} = \{(\tilde{\boldsymbol{x}}_n, \tilde{y}_n)_{n=1,\dots,N}\}$, where $\tilde{\boldsymbol{x}}_n = [\boldsymbol{s}_0^n, \boldsymbol{\theta}^n]$ and $\tilde{y}_n = \sum_{t=0}^{M_n} \gamma^t r(\boldsymbol{s}_t^n, \boldsymbol{\pi}(\boldsymbol{s}_t^n, \boldsymbol{\theta}^n))$. This allows us to directly use (5) and (6) with $\tilde{\mathbb{D}}$ to compute the posterior, while no other changes are made compared to our implementation of Algorithm 1.
- **DDPG:** To illustrate benefits over modern reinforcement learning techniques, we use Deep Deterministic Policy Gradients [65] as an example for the ubiquitous class of actor-critic methods. We use the implementation from [66], which topped the OpenAI Pendulum leaderboard [67]. To ensure fair comparison, we constrain the actor to linear policies with the same parameter bounds as our method.

5.2 Dense Reward Experiments

We firstly compare all methods on the Gymnasium Pendulum environment using the default squared rewards, which provide dense feedback. We slightly modify the environment by defining the angular position and angular velocity as observations, such that the linear controller parameterization is essentially rendered a PD control law [68]. We evaluate all methods based on three different distributions of the initial state: **random**, **upwards**, and **downwards**. When evaluating on the **random** scenario, which corresponds to the default setting of the environment, the initial angular position and velocity are sampled uniformly at random from the entire state space. Thereby, the variance of episode returns is naturally large in this scenario. In order to reduce the randomness due to the initial state distribution, we additionally evaluate the methods in the **upwards** and **downwards** scenarios, where the initial state is sampled from uniform distributions with width 0.2 centered around the upwards and downwards stationary configurations. These settings are highly distinctive from each other: Due to the quadratic rewards, early rewards of roll-outs are relatively flat and optimal returns have small magnitudes in the **upwards** scenario, while the opposite holds for the **downwards** scenario.

This variety in these scenarios highlights crucial differences in the regret of the compared methods as illustrated in Fig. 1. DDPG only exhibits a considerably better performance than random actions in the scenario with random initial states. Since it does not benefit from the exploration provided by the initial states in other scenarios, more than 200 episodes are necessary before the regret decays. In contrast, UCB-based methods do not show this behavior as they inherently ensure sufficient exploration. The Value GP-UCB approach even suffers from the high variance of episode returns in the random scenario. While it also struggles with the flat rewards in the early steps of roll-outs experienced in the upwards scenario, our proposed TDGP-UCB approach shows a consistent performance across all scenarios.

While it would seem reasonable that the advantage of exploiting the temporal difference structure becomes less beneficial with shorter effective horizons, i.e., smaller γ, this is generally not the case. As illustrated in Fig. 2 for the random scenario, the cumulative regret of our TDGP-UCB approach normalized by effective horizon lies consistently below all baseline approaches. It only exhibits a considerable increase when γ is close to 1, which coincides with the performance deterioration

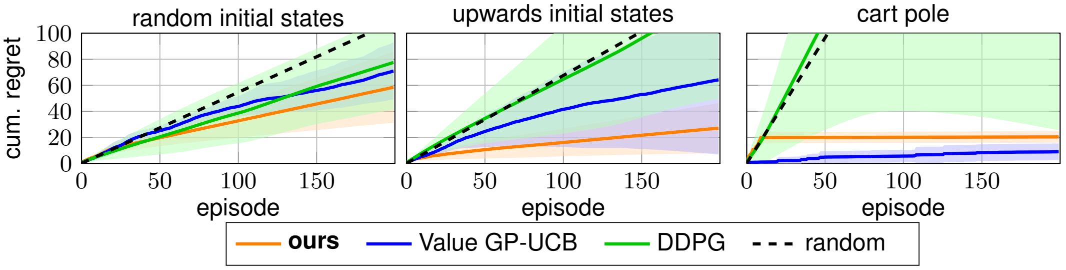

Figure 3: Average cumulative regret for the Gymnasium Pendulum and Cart Pole environments with sparse rewards. Shaded areas illustrate one standard deviation confidence intervals.

of random policies. Hence, the achieved regret of our method strongly correlates with the overall problem difficulty, which further underlines its effectiveness.

5.3 Sparse Reward Experiments

To increase the problem difficulty, we additionally modify the scenarios considered in Section 5.2 by providing only binary rewards, such that we obtain a sparse reward setting. In particular, a reward of 1 is returned if the angular position is within 60 degrees of the vertically upward position and 0 otherwise. Note that the binary rewards in combination with the deterministic pendulum dynamics render the value functions discontinuous. Hence, the experiments in this section constitute an evaluation outside our theoretical assumptions.

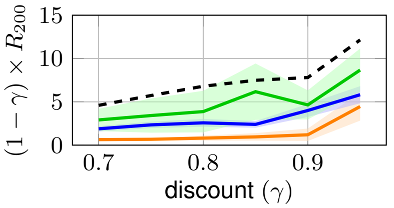

Figure 2: Behavior of cumulative regret normalized by effective horizon depending on the discount factor.

The cumulative regret achieved in the modified random and upwards scenarios is illustrated at the left and center of Fig. 3. Even though the proposed TDGP-UCB algorithm remains to outperform the baseline methods, a considerable increase of the regret can be observed for all methods except DDPG. In particular, the performance of our approach suffers significantly from the high randomness in the random scenario. It should be noted that the apparently linear growth of the cumulative regret is not out of the ordinary: It is known that Bayesian optimization exhibits such a regret in the misspecified setting [69] to which these scenarios correspond to.

While our approach relies on the smoothness of value functions, we can further violate this condition by considering discrete actions. We demonstrate this for the Gymnasium Cart-Pole environment that admits only actions $\{0, 1\}$. We filter the output of the linear policy using the Heavyside step function to enable our approach. To account for the higher-dimensional state space, we additionally do 5 rollouts using random parameters $\boldsymbol{\theta}$, but leave the evaluation setting otherwise unchanged compared to the Pendulum environment. The resulting regret curves are illustrated on the right of Fig. 3. While the performance of our approach is worse than that of the default GP-UCB algorithm, this is completely caused by a steep regret growth before the first hyperparemeter optimization, while the regret remains essentially constant afterwards and exhibits a lower growth than all baseline methods. Therefore, these examples demonstrate the practical effectiveness of our approach beyond its theoretical guarantees.

6 Conclusion and Outlook

In this work, we proposed a novel framework for temporal difference-based learning of value functions with Gaussian process regression. We established regret bounds for UCB-based BO in this framework, for which we analyze the maximum information gain of the temporal difference GP. The effectiveness of our approach is demonstrated in several benchmarks additionally highlighting its practical benefits.

In future work, we aim to address the poor scalability to higher dimensional state and parameter spaces that our approach inherits from BO. For this, we will adopt ideas from local BO [70; 71] to avoid the exploration of the full policy parameter space. Moreover, we will explore approaches to compactly represent neural network policies using concise embeddings referred to as fingerprinting [60], such that our approach becomes applicable to deep RL settings.

Acknowledgments and Disclosure of Funding

This work was supported as a part of NCCR Automation, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 51NF40 225155).

References

[1] Marc Peter Deisenroth. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013. ISSN 1935-8253. doi: 10.1561/2300000021.

[2] Olivier Sigaud and Freek Stulp. Policy search in continuous action domains: an overview. *Neural Networks*, 113:28–40, 2019.

[3] Andreas Doerr, Christian Daniel, Duy Nguyen-tuong, Alonso Marco, Stefan Schaal, Marc Toussaint, and Sebastian Trimpe. Optimizing Long-term Predictions for Model-based Policy Search. In *Proceedings of the Conference on Robot Learning*, pages 1–12, 2017.

[4] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

[5] Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning Dexterous Manipulation for a Soft Robotic Hand from Human Demonstration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3786–3793, 2016.

[6] Sergey Levine and Vladlen Koltun. Guided Policy Search. In *Proceedings of the International Conference on Machine Learning*, 2013.

[7] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104 (1):148–175, 2016.

[8] Roman Garnett. *Bayesian Optimization*. Cambridge University Press, Cambridge, UK, 2023.

[9] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[10] Sayak Ray Chowdhury and Aditya Gopalan. On Kernelized Multi-armed Bandits. In *Proceedings of the International Conference on Machine Learning*, pages 844–853, 2017.

[11] Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. Safe Controller Optimization for Quadrotors with Gaussian Processes. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 491–496, 2016.

[12] Felix Berkenkamp, Andreas Krause, and Angela P. Schoellig. Bayesian Optimization with Safety Constraints: Safe Automatic Parameter Tuning in Robotics. *Machine Learning*, 112(10): 3173–3747, 2023.

[13] Joel A. Paulson, Georgios Makrygiorgos, and Ali Mesbah. Adversarially robust Bayesian optimization for efficient auto-tuning of generic control structures under uncertainty. *AIChE Journal*, 68(6):e17591, June 2022. doi: 10.1002/aic.17591.

[14] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

[15] Sudeep Salgia, Sattar Vakili, and Qing Zhao. A Domain-Shrinking based Bayesian Optimization Algorithm with Order-Optimal Regret Performance. In *Advances in Neural Information Processing Systems*, volume 34, pages 28836–28847. Curran Associates, Inc., 2021.

[16] Zihan Li and Jonathan Scarlett. Gaussian Process Bandit Optimization with Few Batches. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 92–107. PMLR, May 2022.

[17] Sattar Vakili, Nacime Bouziani, Sepehr Jalali, Alberto Bernacchia, and Da-shan Shiu. Optimal Order Simple Regret for Gaussian Process Bandits. In *Advances in Neural Information Processing Systems*, volume 34, pages 21202–21215. Curran Associates, Inc., 2021.

[18] Marcello Fiducioso, Sebastian Curi, Benedikt Schumacher, Markus Gwerder, and Andreas Krause. Safe Contextual Bayesian Optimization for Sustainable Room Temperature PID Control Tuning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5850–5856, Macao, China, August 2019. doi: 10.24963/ijcai.2019/811.

[19] Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic LQR Tuning based on Gaussian Process Global Optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 270–277, 2016.

[20] Farshud Sorourifar, Georgios Makrygirgos, Ali Mesbah, and Joel A. Paulson. A data-driven automatic tuning method for mpc under uncertainty using constrained bayesian optimization. *IFAC-PapersOnLine*, 54(3):243–250, 2021. doi: 10.1016/j.ifacol.2021.08.249. 16th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2021.

[21] R. Calandra, A. Seyfarth, J. Peters, and M. Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, pages 1–19, 2015.

[22] Somil Bansal, Roberto Calandra, Ted Xiao, Sergey Levine, and Claire J. Tomlin. Goal-driven dynamics learning via Bayesian optimization. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 5168–5173, December 2017. doi: 10.1109/CDC.2017.8264425.

[23] Lukas P. Fröhlich, Edgar D. Klenske, Christian G. Daniel, and Melanie N. Zeilinger. Bayesian Optimization for Policy Search in High-Dimensional Systems via Automatic Domain Selection. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 757–764, November 2019. doi: 10.1109/IROS40897.2019.8967736.

[24] Bhavya Sukhija, Lenart Treven, Florian Dörfler, Stelian Coros, and Andreas Krause. NEORL: Efficient Exploration for Nonepisodic RL. In *Advances in Neural Information Processing Systems*, pages 74966–74998, 2024.

[25] Sebastian Curi, Felix Berkenkamp, and Andreas Krause. Efficient Model-Based Reinforcement Learning through Optimistic Policy Search and Planning. In *Advances in Neural Information Processing Systems*, 2020.

[26] Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information Theoretic Regret Bounds for Online Nonlinear Control. In *Advances in Neural Information Processing Systems*, volume 33, pages 15312–15325. Curran Associates, Inc., 2020.

[27] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2017. Publication Title: Trends in Cognitive Sciences.

[28] Aad van der Vaart and Harry van Zanten. Information Rates of Nonparametric Gaussian Process Methods. *Journal of Machine Learning Research*, 12:2095–2119, 2011.

[29] Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences. pages 1–64, 2018. URL `http://arxiv.org/abs/1807.02582`. arXiv: 1807.02582.

[30] K. Hinderer. Lipschitz Continuity of Value Functions in Markovian Decision Processes. *Mathematical Methods of Operations Research*, 62:3–22, 2005.

[31] Hans Harder and Sebastian Peitz. On the continuity and smoothness of the value function in reinforcement learning and optimal control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1935–1940, December 2024. doi: 10.1109/CDC56724.2024.10886529.

[32] Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B Schön. Linearly constrained gaussian processes. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[33] Tadashi Matsumoto and T. J. Sullivan. Images of Gaussian and other stochastic processes under closed, densely-defined, unbounded linear operators. *Analysis and Applications*, 22(3), 2024. doi: 10.1142/S0219530524400025.

[34] Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform Error Bounds for Gaussian Process Regression with Application to Safe Control. In *Advances in Neural Information Processing Systems*, pages 659–669, 2019.

[35] Yasin Abbasi-Yadkori. *Online Learning for Linearly Parametrized Control Problems*. PhD thesis, University of Alberta, Edmonton, 2013.

[36] Peter Auer. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2003.

[37] Johannes Kirschner, Mojmír Mutný, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In *Proceedings of the International Conference on Machine Learning*, pages 5959–5971, 2019.

[38] David Janz, David Burt, and Javier Gonzalez. Bandit optimisation of functions in the Matérn kernel RKHS. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2486–2495. PMLR, June 2020.

[39] Sattar Vakili, Kia Khezeli, and Victor Picheny. On Information Gain and Regret Bounds in Gaussian Process Bandits. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, March 2021.

[40] Andreas Krause and Cheng Ong. Contextual Gaussian Process Bandit Optimization. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

[41] Yaakov Engel, Shie Mannor, and Ron Meir. Bayes Meets Bellman: The Gaussian Process Approach to Temporal Difference Learning. In *Proceedings of the International Conference on Machine Learning*, pages 154–161, 2003.

[42] Devadatta Kulkarni, Darrell Schmidt, and Sze-Kai Tsui. Eigenvalues of tridiagonal pseudo-Toeplitz matrices. *Linear Algebra and its Applications*, 297(1-3):63–80, August 1999. doi: 10.1016/S0024-3795(99)00114-7.

[43] Mohammad Khosravi, Varsha Behrunani, Piotr Myszkorowski, Roy S. Smith, Alisa Rupenyan, and John Lygeros. Performance-Driven Cascade Controller Tuning with Bayesian Optimization. *IEEE Transactions on Industrial Electronics*, 69(1):1032–1042, 2022. doi: 10.1109/TIE.2021.3050356.

[44] Akshara Rai, Rika Antonova, Franziska Meier, and Christopher G Atkeson. Using Simulation to Improve Sample-Efficiency of Bayesian Optimization for Bipedal Robots. *Journal of Machine Learning Research*, 20(49):1–24, 2019.

[45] Benjamin Letham and Eytan Bakshy. Bayesian Optimization for Policy Search via Online-Offline Experimentation. *Journal of Machine Learning Research*, 20(145):1–30, 2019.

[46] Shiming He, Alexander von Rohr, Dominik Baumann, Ji Xiang, and Sebastian Trimpe. Simulation-Aided Policy Tuning for Black-Box Robot Learning. *IEEE Transactions on Robotics*, 41:2533–2548, 2025. doi: 10.1109/TRO.2025.3539192.

[47] Mahdi Kallel, Debabrota Basu, Riad Akrour, and Carlo D'Eramo. Augmented Bayesian Policy Search. In *Proceedings of the International Conference on Learning Representations*, 2024.

[48] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian Reinforcement Learning: A Survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015. doi: 10.1561/2200000049.

[49] Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 201–208, Bonn, Germany, 2005. ACM Press. doi: 10.1145/1102351.1102377.

[50] Robert C Grande, Thomas J Walsh, and Jonathan P How. Sample Efficient Reinforcement Learning with Gaussian Processes. In *Proceedings of the International Conference on Machine Learning*, 2014.

[51] Girish Chowdhary, Miao Liu, Robert Grande, Thomas Walsh, Jonathan How, and Lawrence Carin. Off-policy reinforcement learning with Gaussian processes. *IEEE/CAA Journal of Automatica Sinica*, 1(3):227–238, 2014. doi: 10.1109/JAS.2014.7004680.

[52] Dongruo Zhou, Jiafan He, and Quanquan Gu. Provably Efficient Reinforcement Learning for Discounted MDPs with Feature Mapping. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12793–12802. PMLR, July 2021.

[53] Yuanzhou Chen, Jiafan He, and Quanquan Gu. On the Sample Complexity of Learning Infinite-horizon Discounted Linear Kernel MDPs. In *Proceedings of the 39th International Conference on Machine Learning*, pages 3149–3183. PMLR, June 2022.

[54] Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L. Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 752–759, Helsinki, Finland, 2008. ACM Press. doi: 10.1145/1390156.1390251.

[55] Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael Jordan. Provably Efficient Reinforcement Learning with Kernel and Neural Function Approximations. In *Advances in Neural Information Processing Systems*, volume 33, pages 13903–13916. Curran Associates, Inc., 2020.

[56] Lin Yang and Mengdi Wang. Reinforcement Learning in Feature Space: Matrix Bandit, Kernels, and Regret Bound. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10746–10756. PMLR, November 2020.

[57] Sayak Ray Chowdhury and Rafael Oliveira. Value Function Approximations via Kernel Embeddings for No-Regret Reinforcement Learning. In *Proceedings of The 14th Asian Conference on Machine Learning*, pages 249–264. PMLR, April 2023.

[58] Sing-Yuan Yeh, Fu-Chieh Chang, Chang-Wei Yueh, Pei-Yuan Wu, Alberto Bernacchia, and Sattar Vakili. Sample Complexity of Kernel-Based Q-Learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, pages 453–469. PMLR, April 2023.

[59] Francesco Faccio, Louis Kirsch, and Jürgen Schmidhuber. Parameter-Based Value Functions. In *Proceedings of the International Conference on Learning Representations*, August 2021. doi: 10.48550/arXiv.2006.09226.

[60] Jean Harb, Tom Schaul, Doina Precup, and Pierre-Luc Bacon. Policy Evaluation Networks, February 2020. URL `http://arxiv.org/abs/2002.11833`. arXiv:2002.11833 [cs].

[61] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A Standard Interface for Reinforcement Learning Environments, November 2024. URL `http://arxiv.org/abs/2407.17032`. arXiv:2407.17032 [cs].

[62] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[63] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1–8, 2008.

[64] Christian Fiedler, Carsten W. Scherer, and Sebastian Trimpe. Practical and Rigorous Uncertainty Bounds for Gaussian Process Regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. URL `http://arxiv.org/abs/2105.02796`. arXiv: 2105.02796.

[65] 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 *Proceedings of the International Conference on Learning Representations*, 2016. arXiv: 1509.02971.

[66] Kanishk Navale. Naive multiagent reinforcement learning. `https://github.com/KanishkNavale/Naive-MultiAgent-ReinforcementLearning`, 2022.

[67] Openai gym leaderboard, pendulum. `https://github.com/openai/gym/wiki/Leaderboard#pendulum-v0`. Accessed: 28-08-2025.

[68] M Sami Fadali and Antonio Visioli. *Digital Control Engineering Analysis and Design Second Edition*. Academic Press, Waltham, MA, 2012.

[69] Ilija Bogunovic and Andreas Krause. Misspecified Gaussian Process Bandit Optimization. In *Advances in Neural Information Processing Systems*, 2021.

[70] Sarah Müller, Alexander von Rohr, and Sebastian Trimpe. Local policy search with Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 34, pages 20708–20720. Curran Associates, Inc., 2021.

[71] Quan Nguyen, Kaiwen Wu, Jacob R Gardner, and Roman Garnett. Local Bayesian optimization via maximizing probability of descent. In *Advances in Neural Information Processing Systems*, pages 13190–13202, 2022.

[72] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science+Business Media, New York, NY, 2008.

A Error bounds for Gaussian process regression

Lemma 1. *Consider a value function V satisfying Assumption 2. Then, V is L_V-Lipschitz continuous with $L_V = \sqrt{L_k}B$.*

Proof. It follows from [72, Corollary 4.36] that a function with bounded RKHS norm satisfies

$$\left|\frac{\partial}{\partial s_i}V(\boldsymbol{s},\boldsymbol{\theta})\right|^2 \leq B\frac{\partial^2}{\partial s_i\partial s_i'}k([\boldsymbol{s};\boldsymbol{\theta}],[\boldsymbol{s}';\boldsymbol{\theta}])\Big|_{\boldsymbol{s}'=\boldsymbol{s}}. \tag{22}$$

By defining the Lipschitz constant via the maximum derivative of V, we consequently obtain

$$\begin{align}
L_V &\leq \max_{\boldsymbol{s}\in\mathcal{S},\boldsymbol{\theta}\in\boldsymbol{\Theta}} \|\nabla_{\boldsymbol{s}}V(\boldsymbol{s},\boldsymbol{\theta})\| \tag{23}\\
&\leq \max_{\boldsymbol{s}\in\mathcal{S},\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left\|\begin{bmatrix}\frac{\partial}{\partial s_1}V(\boldsymbol{s},\boldsymbol{\theta})\\ \vdots \\ \frac{\partial}{\partial s_{d_s}}V(\boldsymbol{s},\boldsymbol{\theta})\end{bmatrix}\right\| \tag{24}\\
&\leq B \max_{\boldsymbol{s}\in\mathcal{S},\boldsymbol{\theta}\in\boldsymbol{\Theta}} \left\|\begin{bmatrix}\sqrt{\frac{\partial^2}{\partial s_1\partial s_1'}k([\boldsymbol{s};\boldsymbol{\theta}],[\boldsymbol{s}';\boldsymbol{\theta}])\Big|_{\boldsymbol{s}'=\boldsymbol{s}}}\\ \vdots \\ \sqrt{\frac{\partial^2}{\partial s_{d_s}\partial s_{d_s}'}k([\boldsymbol{s};\boldsymbol{\theta}],[\boldsymbol{s}';\boldsymbol{\theta}])\Big|_{\boldsymbol{s}'=\boldsymbol{s}}}\end{bmatrix}\right\| \tag{25}\\
&\leq B \max_{\boldsymbol{s}\in\mathcal{S},\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sqrt{\sum_{i=1}^{d_s}\frac{\partial^2}{\partial s_i\partial s_i'}k([\boldsymbol{s};\boldsymbol{\theta}],[\boldsymbol{s}';\boldsymbol{\theta}])\Big|_{\boldsymbol{s}'=\boldsymbol{s}}} \tag{26}\\
&\leq B \max_{\boldsymbol{s}\in\mathcal{S},\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sqrt{\operatorname{trace}\left(\nabla_{\boldsymbol{s}}\nabla_{\boldsymbol{s}'}^T k([\boldsymbol{s};\boldsymbol{\theta}],[\boldsymbol{s}';\boldsymbol{\theta}])\Big|_{\boldsymbol{s}'=\boldsymbol{s}}\right)}. \tag{27}
\end{align}$$

Finally, the result follows by bounding the trace using L_k. □

Lemma 2. *Consider a L_V-Lipschitz value function V and process noise satisfying Assumption 1. Then, $\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\boldsymbol{\theta}) - r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$ is $2\sqrt{d_s}L_V\lambda$-sub-Gaussian.*

Proof. It directly follows from the definition of ΔV that

$$\mathbb{E}_{\boldsymbol{w}}[\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\boldsymbol{\theta})] = r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta})).$$

Moreover, it is straightforward to see that all the randomness of $\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\boldsymbol{\theta})$ appears in

$$V(\boldsymbol{s}^+,\boldsymbol{\theta}) = V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta})$$

with $\tilde{\boldsymbol{s}}^+ = \mathbb{E}_{\boldsymbol{s}^+}[\boldsymbol{s}^+]$ for $\boldsymbol{s}^+ \sim p(\boldsymbol{s}^+|\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$ and $\boldsymbol{w} = \boldsymbol{s}^+ - \tilde{\boldsymbol{s}}^+$. Therefore, it suffices to show

$$\mathbb{E}_{\boldsymbol{w}}\left[\exp\left(\frac{(V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta}) - \mathbb{E}_{\boldsymbol{w}}[V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta})])^2}{4d_sL_V^2\lambda^2}\right)\right] \leq 2 \tag{28}$$

to prove that $\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\boldsymbol{\theta}) - r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$ is $2\sqrt{d_s}L_V\lambda$-sub-Gaussian. For this purpose, let $\boldsymbol{w}'$ be an independent copy of $\boldsymbol{w}$. Since $\boldsymbol{w}'$ has the same distribution as $\boldsymbol{w}$, we can reformulate the left side of (28) as

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{w}}\left[\exp\left(\frac{(V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta}) - \mathbb{E}_{\boldsymbol{w}}[V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta})])^2}{4d_sL_V^2\lambda^2}\right)\right] = \\
&\qquad\qquad \mathbb{E}_{\boldsymbol{w}}\left[\exp\left(\frac{(V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta}) - \mathbb{E}_{\boldsymbol{w}'}[V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w}',\boldsymbol{\theta})])^2}{4d_sL_V^2\lambda^2}\right)\right].
\end{aligned}$$

Using Jensen's inequality, we can pull the expectation out of the exponential yielding

$$\begin{aligned}
&\exp\left(\frac{(V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta}) - \mathbb{E}_{\boldsymbol{w}}[V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta})])^2}{4d_sL_V^2\lambda^2}\right) \leq \\
&\qquad\qquad \mathbb{E}_{\boldsymbol{w}'}\left[\exp\left(\frac{(V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta}) - V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w}',\boldsymbol{\theta}))^2}{4d_sL_V^2\lambda^2}\right)\right].
\end{aligned}$$

This allows us to exploit the Lipschitz continuity of V via

$$\mathbb{E}_{\boldsymbol{w},\boldsymbol{w}'}\left[\exp\left(\frac{(V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w},\boldsymbol{\theta}) - V(\tilde{\boldsymbol{s}}^+ + \boldsymbol{w}',\boldsymbol{\theta}))^2}{4d_sL_V^2\lambda^2}\right)\right] \leq \mathbb{E}_{\boldsymbol{w},\boldsymbol{w}'}\left[\exp\left(\frac{(L_V(\|\boldsymbol{w}\| + \|\boldsymbol{w}'\|))^2}{4d_sL_V^2\lambda^2}\right)\right].$$

Due to Young's inequality we can bound this expression in terms of $\|\boldsymbol{w}\|^2$ and $\|\boldsymbol{w}'\|^2$, i.e.,

$$\mathbb{E}_{\boldsymbol{w},\boldsymbol{w}'}\left[\exp\left(\frac{(L_V(\|\boldsymbol{w}\| + \|\boldsymbol{w}'\|))^2}{4d_sL_V^2\lambda^2}\right)\right] \leq \mathbb{E}_{\boldsymbol{w},\boldsymbol{w}'}\left[\exp\left(\frac{2L_V^2(\|\boldsymbol{w}\|^2 + \|\boldsymbol{w}'\|^2)}{4d_sL_V^2\lambda^2}\right)\right].$$

The independence of $\boldsymbol{w}$ and $\boldsymbol{w}'$ and their identical distributions ensures that

$$\mathbb{E}_{\boldsymbol{w},\boldsymbol{w}'}\left[\exp\left(\frac{2L_V^2(\|\boldsymbol{w}\|^2 + \|\boldsymbol{w}'\|^2)}{4d_sL_V^2\lambda^2}\right)\right] = \mathbb{E}_{\boldsymbol{w}}\left[\exp\left(\frac{4d_sL_V^2\|\boldsymbol{w}\|^2)}{4L_V^2\lambda^2}\right)\right].$$

Finally, let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{d_x}$ be an orthonormal basis for $\mathbb{R}^{d_s}$. Then, we have

$$\begin{aligned}\|\boldsymbol{w}\|^2 &= \boldsymbol{w}^T\left(\sum_{i=1}^{d_s}\boldsymbol{b}_i\boldsymbol{b}_i^T\right)\boldsymbol{w}\\ &\leq d_s \max_{i=1,\ldots,d_x} \boldsymbol{w}^T\boldsymbol{b}_i\boldsymbol{b}_i^T\boldsymbol{w}\\ &\leq d_s \sup_{\boldsymbol{v}:\|\boldsymbol{v}\|=1} \boldsymbol{w}^T\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{w}\\ &\leq d_s \sup_{\boldsymbol{v}:\|\boldsymbol{v}\|=1} (\boldsymbol{v}^T\boldsymbol{w})^2.\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}\mathbb{E}_{\boldsymbol{w}}\left[\exp\left(\frac{4L_V^2\|\boldsymbol{w}\|^2)}{4d_sL_V^2\lambda^2}\right)\right] &\leq \sup_{\boldsymbol{v}:\|\boldsymbol{v}\|=1}\mathbb{E}_{\boldsymbol{w}}\left[\exp\left(\frac{4d_sL_V^2(\boldsymbol{v}^T\boldsymbol{w})^2)}{4d_sL_V^2\lambda^2}\right)\right]\\ &\leq \sup_{\boldsymbol{v}:\|\boldsymbol{v}\|=1}\mathbb{E}_{\boldsymbol{w}}\left[\exp\left(\frac{(\boldsymbol{v}^T\boldsymbol{w})^2)}{\lambda^2}\right)\right]\\ &\leq 2,\end{aligned}$$

where the last inequality is a direct consequence of Assumption 1. Hence, (28) is satisfied, which concludes the proof. □

Proposition 1. *Consider a stochastic dynamical system* (1) *satisfying Assumption 1, assume that the value function V satisfies Assumption 2, and let $m_V \equiv 0$. Then, the error for learning V from a data set $\mathbb{D}^n$ using Gaussian process regression with posterior mean* (11) *and variance* (12) *is bounded by*

$$|\mu_V(\boldsymbol{x}) - V(\boldsymbol{x})| \leq \beta(\delta)\sigma_V(\boldsymbol{x}) \tag{29}$$

jointly for all $\boldsymbol{x},\boldsymbol{\theta}$ in a compact domain $\mathcal{S}\times\Theta$ with probability $1-\delta$, $\delta \in (0,1)$, *where*

$$\beta(\delta) = 2\sqrt{d_sL_k}B\lambda\sqrt{\log\det\left(\boldsymbol{I} + \frac{1}{\sigma^2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)\right) - 2\log(\delta)} + \sigma B. \tag{30}$$

Proof. The target values for the GP model with prior mean $m \equiv 0$ and kernel (9) are of the form $\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\boldsymbol{\theta})$ but we use labels $r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$ for computing the posterior mean (11). Thus, the labels are perturbed by noise $\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\boldsymbol{\theta}) - r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$. Note that the value function V is L_V-Lipschitz with $L_V = \sqrt{L_k}B$ due to Lemma 1. Therefore, it follows from Lemma 2 that $\Delta V(\boldsymbol{s},\boldsymbol{s}^+,\boldsymbol{\theta}) - r(\boldsymbol{s},\boldsymbol{\pi}(\boldsymbol{s},\boldsymbol{\theta}))$ is $2\sqrt{d_sL_k}B\lambda$-sub-Gaussian. Finally, the sub-Gaussianity of the noise in GP regression allows the application of [35, Theorem 3.11], which guarantees a learning error bound (29) with β defined in (30). □

B Regret Bounds for Optimistic Optimization

Lemma 3. *Consider a dynamical system* (1) *with process noise satisfying Assumption 1 and assume that the value function V satisfies Assumption 2. If $\boldsymbol{\theta}_{\mathrm{UCB}}$ is chosen according to* (16) *it holds with probability $1-\delta$ that*

$$J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_{\mathrm{UCB}}) \leq 2\beta \mathbb{E}_{\boldsymbol{s}_0}\left[\sigma_V\left([\boldsymbol{s}_0; \boldsymbol{\theta}_{\mathrm{UCB}}]\right)\right]. \tag{31}$$

Proof. Due to Proposition 1, $\hat{J}$ upper bounds J with probability $1-\delta$. Thus, we have

$$J(\boldsymbol{\theta}^*) \leq \hat{J}(\boldsymbol{\theta}^*)$$

with probability $1-\delta$. Moreover, (16) implies that

$$\hat{J}(\boldsymbol{\theta}^*) \leq \hat{J}(\boldsymbol{\theta}_{\mathrm{UCB}}).$$

Combining these two inequalities, we obtain

$$J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_{\mathrm{UCB}}) \leq \hat{J}(\boldsymbol{\theta}_{\mathrm{UCB}}) - J(\boldsymbol{\theta}_{\mathrm{UCB}})$$

with probability $1-\delta$. Exploiting the linearity of the expectation and the definition of $\hat{J}$ yields

$$J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_{\mathrm{UCB}}) \leq \mathbb{E}_{\boldsymbol{s}_0}\left[\mu_V\left([\boldsymbol{s}_0; \boldsymbol{\theta}_{\mathrm{UCB}}]\right) + \beta\sigma_V\left([\boldsymbol{s}_0; \boldsymbol{\theta}_{\mathrm{UCB}}]\right) - V(\boldsymbol{s}_0, \boldsymbol{\theta}_{\mathrm{UCB}})\right]$$

with probability $1-\delta$. Finally, we can bound $|\mu_V\left([\boldsymbol{s}_0; \boldsymbol{\theta}_{\mathrm{UCB}}]\right) - V(\boldsymbol{s}_0, \boldsymbol{\theta}_{\mathrm{UCB}})| \leq \beta^n \sigma_V\left([\boldsymbol{s}_0; \boldsymbol{\theta}_{\mathrm{UCB}}]\right)$ with probability $1-\delta$ by employing Proposition 1 once more, which immediately implies (31).

□

Lemma 4. *Let $\tau^n = \{(\boldsymbol{s}_m, r_m)_{m=1,\ldots,M}\}$, $\boldsymbol{x}_m = [\boldsymbol{s}_m, \boldsymbol{\theta}]$, and $\boldsymbol{z}_m = [\boldsymbol{x}_m, \boldsymbol{x}_{m+1}]$. Then, the posterior variance* (12) *satisfies*

$$\sigma_V^2(\boldsymbol{x}_0) \leq \frac{(M+1)\gamma^M}{2}\sigma_V^2(\boldsymbol{x}_M) + \frac{\gamma^M - \gamma^{M+1} + 2}{2(1-\gamma)}\sum_{m=0}^{M-1}\gamma^m \sigma_\Delta^2(\boldsymbol{z}_m) \tag{32}$$

for σ_Δ^2 defined in (17).

Proof. Due to the Bayesian foundations of GPs, we know that $\sigma_V^2(\boldsymbol{x}_0)$ is the conditional variance of $V(\boldsymbol{x}_0)$ under the GP prior, i.e.,

$$\sigma_V^2(\boldsymbol{x}_0) = \mathbb{V}\left[V(\boldsymbol{x}_0)|\mathbb{D}\right]. \tag{33}$$

For notational simplicity, we drop the conditioning on data for notational simplicity in the following derivations and simply write $\sigma_V^2(\boldsymbol{x}_0) = \mathbb{V}\left[V(\boldsymbol{x}_0)\right]$. Observe that we can expand the value function in terms of the temporal difference via

$$\begin{aligned} V(\boldsymbol{x}_m) &= V(\boldsymbol{x}_m) - \gamma V(\boldsymbol{x}_{m+1}) + \gamma V(\boldsymbol{x}_{m+1}), \\ &= \Delta V(\boldsymbol{z}_m) + \gamma V(\boldsymbol{x}_{m+1}). \end{aligned}$$

Employing this identity recursively, we obtain

$$V(\boldsymbol{x}_0) = \gamma^M V(\boldsymbol{x}_M) + \sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m).$$

Therefore, we can express the GP variance $\sigma_V^2(\boldsymbol{x}_0)$ as

$$\sigma_V^2(\boldsymbol{x}_0) = \mathbb{V}\left[\gamma^M V(\boldsymbol{x}_M) + \sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m)\right],$$

which we can equivalently express as

$$\begin{aligned} \sigma_V^2(\boldsymbol{x}_0) =& \mathrm{Cov}\left(\gamma^M V(\boldsymbol{x}_M), \gamma^M V(\boldsymbol{x}_M)\right) + 2\mathrm{Cov}\left(\gamma^M V(\boldsymbol{x}_M), \sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m)\right) \\ &+ \mathrm{Cov}\left(\sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m), \sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m)\right). \end{aligned} \tag{34}$$

For the first term, linearity of the covariance in each argument together with the definition of $\sigma_V^2(\boldsymbol{x}_M)$ yields

$$\mathrm{Cov}\left(\gamma^M V(\boldsymbol{x}_M), \gamma^M V(\boldsymbol{x}_M)\right) = \gamma^{2M}\sigma_V^2(\boldsymbol{x}_M). \tag{35}$$

For the second term, we obtain

$$\begin{aligned}
\mathrm{Cov}\left(\gamma^M V(\boldsymbol{x}_M), \sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m)\right) &= \sum_{m=0}^{M-1}\gamma^{M+m}\mathrm{Cov}\left(V(\boldsymbol{x}_M), \Delta V(\boldsymbol{z}_m)\right) \\
&\leq \sum_{m=0}^{M-1}\gamma^{M+m}\sqrt{\mathbb{V}[V(\boldsymbol{x}_M)]}\sqrt{\mathbb{V}[\Delta V(\boldsymbol{z}_m)]} \\
&\leq \frac{1}{2}\sum_{m=0}^{M-1}\gamma^{M+m}\left(\mathbb{V}[V(\boldsymbol{x}_M)] + \mathbb{V}[\Delta V(\boldsymbol{z}_m)]\right) \\
&\leq \frac{1}{2}\sum_{m=0}^{M-1}\gamma^{M+m}\left(\sigma_V^2(\boldsymbol{x}_M) + \sigma_\Delta^2(\boldsymbol{z}_m)\right) \\
&\leq \frac{M\gamma^M}{2}\sigma_V^2(\boldsymbol{x}_M) + \frac{1}{2}\sum_{m=0}^{M-1}\gamma^{M+m}\sigma_\Delta^2(\boldsymbol{z}_m). \qquad (36)
\end{aligned}$$

The first line follows from the linearity of the covariance in each argument, the second line is due to the Cauchy-Schwarz inequality, Young's inequality results in the third line, the fourth line is a consequence of the definitions of $\sigma_\Delta^2(\boldsymbol{x}_m)$ and $\sigma_V^2(\boldsymbol{x}_M)$, and the last line uses $\gamma^{M+m} \leq \gamma^M$. Similarly, we obtain for the last term in (34) that

$$\begin{aligned}
\mathrm{Cov}\left(\sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m), \sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m)\right) &= \sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\gamma^{m+m'}\mathrm{Cov}\left(\Delta V(\boldsymbol{z}_m), \Delta V(\boldsymbol{z}_{m'})\right) \\
&\leq \sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\gamma^{m+m'}\sqrt{\mathbb{V}[\Delta V(\boldsymbol{z}_m)]}\sqrt{\mathbb{V}[\Delta V(\boldsymbol{z}_{m'})]} \\
&\leq \frac{1}{2}\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\gamma^{m+m'}\left(\mathbb{V}[\Delta V(\boldsymbol{z}_m)] + \mathbb{V}[\Delta V(\boldsymbol{z}_{m'})]\right)
\end{aligned}$$

using the linearity of the covariance in both arguments in the first line, the Cauchy-Schwarz inequality in the second line, and Young's inequality in the last line. Observe that

$$\begin{aligned}
\sum_{m=0}^{M-1}\sum_{m'=0}^{M-1}\gamma^{m+m'}\mathbb{V}[\Delta V(\boldsymbol{z}_m)] &= \sum_{m=0}^{M-1}\gamma^m\mathbb{V}[\Delta V(\boldsymbol{z}_m)]\sum_{m'=0}^{M-1}\gamma^{m'} \\
&\leq \frac{1}{1-\gamma}\sum_{m=0}^{M-1}\gamma^m\mathbb{V}[\Delta V(\boldsymbol{z}_m)] \\
&\leq \frac{1}{1-\gamma}\sum_{m=0}^{M-1}\gamma^m\sigma_\Delta^2(\boldsymbol{z}_m)
\end{aligned}$$

where the second line is due to $\sum_{m'=0}^{M-1}\gamma^{m'} \leq \sum_{m'=0}^{\infty}\gamma^{m'} = \frac{1}{1-\gamma}$ and the third line follows from the definition of $\sigma_\Delta^2(\boldsymbol{x}_m)$. Therefore, we have

$$\mathrm{Cov}\left(\sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m), \sum_{m=0}^{M-1}\gamma^m \Delta V(\boldsymbol{z}_m)\right) \leq \frac{1}{1-\gamma}\sum_{m=0}^{M-1}\gamma^m\sigma_\Delta^2(\boldsymbol{z}_m). \tag{37}$$

Substituting (35), (36), (37) into (34), we obtain

$$\sigma_V^2(\boldsymbol{x}_0) \leq \gamma^{2M}\sigma_V^2(\boldsymbol{x}_M) + \frac{M\gamma^M}{2}\sigma_V^2(\boldsymbol{x}_M) + \frac{1}{2}\sum_{m=0}^{M-1}\gamma^{M+m}\sigma_\Delta^2(\boldsymbol{z}_m) + \frac{1}{1-\gamma}\sum_{m=0}^{M-1}\gamma^m\sigma_\Delta^2(\boldsymbol{z}_m).$$

Finally, we have $\gamma^{2M} \leq \gamma^M$, such that we can simplify this expression to

$$\sigma_V^2(\boldsymbol{x}_0) \leq \frac{(M+1)\gamma^M}{2}\sigma_V^2(\boldsymbol{x}_M) + \frac{\gamma^M - \gamma^{M+1} + 2}{2(1-\gamma)} \sum_{m=0}^{M-1} \gamma^m \sigma_\Delta^2(\boldsymbol{z}_m),$$

which concludes the proof.

□

Lemma 5. *Consider a sequence of training input sets $\mathbb{X}_n = \{(\boldsymbol{x}_i)_{i=1,\dots,n}\}$ with $\mathbb{X}_n \subset \mathbb{X}_{n+1}$. Let $\sigma_{g,n}^2$ be defined by* (6) *with training input set $\mathbb{X}_n$ and prior $\mathcal{GP}(m_g, k_g)$. Moreover, define $[\boldsymbol{K}_g(\boldsymbol{X}_N)]_{i,j} = k_g(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Then, it holds that*

$$\log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_g(\boldsymbol{X}_N)) = \sum_{n=1}^{N} \log(1 + \sigma^{-2}\sigma_{g,n-1}^2(\boldsymbol{x}_n)). \tag{38}$$

Proof. Let $y_n = g(\boldsymbol{x}_n) + \epsilon_n$ for an arbitrary function $g \sim \mathcal{N}(m_g, k_g)$ and $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. noise. Then, we have $\boldsymbol{y}_N | \boldsymbol{g}_N \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$ and $\boldsymbol{y}_N \sim \mathcal{N}(\boldsymbol{m}_g, \boldsymbol{K}_{\boldsymbol{g}_N} + \sigma^2\boldsymbol{I})$, where $[\boldsymbol{y}_N]_i = y_i$, $[\boldsymbol{g}_N]_i = g(\boldsymbol{x}_i)$ and $[\boldsymbol{m}_g]_i = m_g(\boldsymbol{x}_i)$. Therefore, it follows that

$$\frac{1}{2}\log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_g(\boldsymbol{X}_N)) = H(\boldsymbol{y}_N) - H(\boldsymbol{y}_N|\boldsymbol{g}_N),$$

where H is the differential entropy. Due to the chain rule for differential entropies, we have

$$H(\boldsymbol{y}_N) = \sum_{n=1}^{N-1} H(y_{n+1}|\boldsymbol{y}_n) + H(y_1).$$

Moreover, it holds that $y_n|\boldsymbol{y}_{n-1} \sim \mathcal{N}(\mu_{g,n-1}(\boldsymbol{x}_n), \sigma_{g,n-1}^2(\boldsymbol{x}_n))$ such that

$$\sum_{n=1}^{N-1} H(y_{n+1}|\boldsymbol{y}_n) + H(y_1) = \frac{1}{2}\sum_{n=1}^{N} \log(\sigma^2 + \sigma_{g,n-1}^2(\boldsymbol{x}_n)).$$

Combining these identities, we obtain (38), which concludes the proof. □

Lemma 6. *Let M_n be a sequence such that $\gamma^{M_n} \leq {}^{\kappa}\!/_{n^2}$ for $\kappa \in (0,1)$. Then,*

$$\sum_{n=1}^{\infty} \frac{(M_n+1)\gamma^{M_n}}{2} \leq c_1 \tag{39}$$

with

$$c_1 = \frac{\kappa\pi^2(\log(\kappa) + e\log(\gamma) - 1) - 12\kappa e}{12e\log(\gamma)}. \tag{40}$$

Proof. Since $\gamma^{M_n} \leq {}^{\kappa}\!/_{n^2}$, we have

$$\begin{aligned}\sum_{n=1}^{\infty} \frac{\gamma^{M_n}}{2} &\leq \sum_{n=1}^{\infty} \frac{\kappa}{2n^2} \\ &\leq \frac{\kappa\pi^2}{12}.\end{aligned} \tag{41}$$

Moreover, $\gamma^{M_n} \leq {}^{\kappa}\!/_{n^2}$ implies $M_n \geq \log(\kappa/n^2)/\log(\gamma)$. Let $\alpha(n) = M_n - \log(\kappa/n^2)/\log(\gamma)$. Therefore, we obtain

$$\begin{aligned}\sum_{n=1}^{\infty} \frac{M_n\gamma^{M_n}}{2} &= \sum_{n=1}^{\infty} \frac{(\frac{\log(\kappa/n^2)}{\log(\gamma)} + \alpha(n))\frac{\kappa}{n^2}\gamma^{\alpha(n)}}{2}, \\ &\leq \kappa\sum_{n=1}^{\infty} \frac{\log(\kappa/n^2)}{2\log(\gamma)n^2} + \frac{\alpha(n)\gamma^{\alpha(n)}}{2n^2},\end{aligned}$$

where the second line follows from the fact that $\gamma^{\alpha(n)} \leq 1$. The second term can be bounded using

$$\max_{q\in\mathbb{R}} q\gamma^q \leq -\frac{1}{e\log(\gamma)}$$

which yields

$$\sum_{n=1}^{\infty} \frac{M_n\gamma^{M_n}}{2} \leq \frac{\kappa}{2\log(\gamma)} \sum_{n=1}^{\infty} \frac{\log(\kappa) - 2\log(n)}{n^2} - \frac{1}{en^2}.$$

Furthermore, it can be shown that $\sum_{n=1}^{\infty} \log(n)/n^2 \leq 1$, such that

$$\begin{aligned}\sum_{n=1}^{\infty} \frac{M_n\gamma^{M_n}}{2} &\leq \frac{\kappa}{2\log(\gamma)} \left(\frac{\pi^2\log(\kappa)}{6} - 2 - \frac{\pi^2}{6e}\right)\\ &\leq \frac{\kappa\pi^2\log(\kappa) - 12\kappa e - \kappa\pi^2}{12e\log(\gamma)}. \end{aligned} \tag{42}$$

Combining (41) and (42) concludes the proof. □

Theorem 1. *Consider a dynamical system* (1) *satisfying Assumption 1, such that Assumption 2 holds for the value function V. Assume that parameters $\boldsymbol{\theta}_{\mathrm{UCB}}^n$ are chosen via* (16) *with GP mean* (11) *and variance* (12) *learned from a data set $\mathbb{D}^n$ which is obtained from roll-outs of length $\gamma^{M_n} \leq {}^{\kappa}/_{n^2}$ for $\kappa \in (0,1)$ in each episode $n = 1, \ldots, N$. Then, the regret* (3) *satisfies*

$$R_N \leq c\sqrt{NM_N\Gamma_{k_\Delta}(NM_N)(\Gamma_{k_\Delta}(NM_N) + \log(N))} \tag{43}$$

with probability $1-\delta$ and constant $c \in \mathbb{R}_{\geq 0}$ for $N > 1$.

Proof. Since we define β_n using $\delta_n = {}^{6\delta}/_{\pi^2 n^2}$ in every episode, Lemma 3 together with the union bound yields

$$\sum_{n=1}^{N} J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_{\mathrm{UCB}}^n) \leq \sum_{n=1}^{N} 2\beta_{n-1}\mathbb{E}_{\boldsymbol{s}_0}\left[\sigma_{V,n-1}\left([\boldsymbol{s}_0;\boldsymbol{\theta}_{\mathrm{UCB}}^n]\right)\right], \tag{44}$$

$$\leq \mathbb{E}_{\boldsymbol{s}_0}\left[\sum_{n=1}^{N} 2\beta_{n-1}\sigma_{V,n-1}\left([\boldsymbol{s}_0;\boldsymbol{\theta}_{\mathrm{UCB}}^n]\right)\right], \tag{45}$$

where the second line follows from the linearity of the expectation. We expand the variances using Lemma 4, such that we obtain

$$\begin{aligned}\sum_{n=1}^{N} J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_{\mathrm{UCB}}^n) \leq &\mathbb{E}_{\boldsymbol{s}_0}\left[\sum_{n=1}^{N} \beta_{n-1}\left(2\frac{(M_n+1)\gamma^{M_n}}{2}\sigma_{V,n-1}^2(\boldsymbol{x}_{M_n}^n)\right.\right.\\ &\left.\left.+\frac{\gamma^{M_n} - \gamma^{M_n+1} + 2}{2(1-\gamma)}\sum_{m=0}^{M_n-1}\gamma^m\sigma_{\Delta,n-1}^2(\boldsymbol{z}_m^n)\right)^{\frac{1}{2}}\right]\end{aligned} \tag{46}$$

with probability $1-\delta$ for $\boldsymbol{x}_0^n = [\boldsymbol{s}_0;\boldsymbol{\theta}_{\mathrm{UCB}}^n]$. Due to the Cauchy-Schwarz inequality, we can change the order between summation and square root resulting in

$$\begin{aligned}\sum_{n=1}^{N} J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_{\mathrm{UCB}}^n) \leq &\mathbb{E}_{\boldsymbol{s}_0}\left[\left(N\sum_{n=1}^{N} 2\beta_{n-1}^2\frac{(M_n+1)\gamma^{M_n}}{2}\sigma_{V,n-1}^2(\boldsymbol{x}_{M_n}^n)\right.\right.\\ &\left.\left.+\frac{\gamma^{M_n} - \gamma^{M_n+1} + 2}{2(1-\gamma)}\sum_{m=0}^{M_n-1}\gamma^m\beta_{n-1}^2\sigma_{\Delta,n-1}^2(\boldsymbol{z}_m^n)\right)^{\frac{1}{2}}\right]\end{aligned} \tag{47}$$

with probability $1-\delta$. Here, the index $n-1$ in $\sigma_{V,n-1}^2$ and $\sigma_{\Delta,n-1}^2$ indicates that these are the posterior variances given the data set $\mathbb{D}^{n-1}$. We continue to analyze the two terms on the right side of this inequality separately. For the first term, observe that continuity of the kernel k_V guarantees continuity of σ_V^2, such that the compact set $\mathcal{S}$ implies the existence of a finite value $\max_{\boldsymbol{x}\in\mathcal{X}} \sigma^2_{V,0}(\boldsymbol{x}) \geq \sigma^2_{V,n}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X} = \mathcal{S} \times \Theta$. Therefore, Lemma 6 ensures

$$\sum_{n=1}^{N} 2\beta_{n-1}^2 \frac{(M_n+1)\gamma^{M_n}}{2} \sigma^2_{V,n-1}(\boldsymbol{x}^n_{M_n}) \leq 2\beta_N^2 c_1 \max_{\boldsymbol{x}\in\mathcal{X}} \sigma^2_{V,0}(\boldsymbol{x}). \tag{48}$$

For the second term in (47), we first consider the sum over the sub-sequence $\boldsymbol{z}_0^n$. Moreover, let $\sigma^2_{\Delta,i-1}$ denote the variance based on training inputs $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{i-1}$ for $\boldsymbol{z}_i = \boldsymbol{x}^{n_i}_{m_i}$ with $n_i = \min_{\sum_{k=1}^{n'} M_k \geq i} n'$ and $m_i = i - 1 - \sum_{k=1}^{n_i-1} M_k$, i.e., the first $i-1$ samples in the data set. Then, it is straightforward to see that we can bound sum over the sub-sequence $\boldsymbol{z}_0^n$ by

$$\sum_{n=1}^{N} \sigma^2_{\Delta,n-1}(\boldsymbol{z}_0^n) \leq \sum_{i=1}^{\sum_{n=1}^{N} M_n} \sigma^2_{\Delta,i-1}(\boldsymbol{z}_0^n), \tag{49}$$

since $\sigma^2_{\Delta,n-1}(\boldsymbol{z}_0^n) = \sigma^2_{\Delta,i-1}(\boldsymbol{z}_0^n)$ for $i = (n-1)M_n + 1$. By permuting the order of training samples, this is analogously possible for all other sub-sequences $\boldsymbol{z}_i^n$, such that

$$\sum_{n=1}^{N} \sum_{m=0}^{M_n-1} \gamma^m \beta^2_{n-1} \sigma^2_{\Delta,n-1}(\boldsymbol{z}^n_m) \leq M_N \beta_N^2 \sum_{i=1}^{\sum_{n=1}^{N} M_n} \sigma^2_{\Delta,i-1}(\boldsymbol{z}_i) \tag{50}$$

since $\gamma \leq 1$. We can upper bound the variances through through logarithmic terms via

$$\sigma^2_{\Delta,i-1}(\boldsymbol{z}_i) \leq \sigma^2 \frac{\sigma^{-2}\sigma^2_{\Delta,i-1}(\boldsymbol{z}_i)}{\log(1+\sigma^{-2}\sigma^2_{\Delta,i-1}(\boldsymbol{z}_i))} \log(1+\sigma^{-2}\sigma^2_{\Delta,i-1}(\boldsymbol{z}_i)) \tag{51}$$

$$\leq c_2 \log(1 + \sigma^{-2}\sigma^2_{\Delta,i-1}(\boldsymbol{z}_i)) \tag{52}$$

where the second line follows from the monotonous growth of $\frac{q}{\log(1+q)}$, such that

$$c_2 = \frac{\max_{\boldsymbol{z}\in\mathcal{Z}} \sigma^2_{\Delta,0}(\boldsymbol{z})}{\log(1 + \sigma^{-2} \max_{\boldsymbol{z}\in\mathcal{Z}} \sigma^2_{\Delta,0}(\boldsymbol{z}))}. \tag{53}$$

This allows us to apply Lemma 5 to obtain

$$\sum_{n=1}^{N} \sum_{m=0}^{M_n-1} \gamma^m \beta^2_{n-1} \sigma^2_{\Delta,n-1}(\boldsymbol{z}^n_m) \leq c_2 M_N \beta_N^2 \log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)). \tag{54}$$

Substituting (48) and (54) into (47) results in

$$\sum_{n=1}^{N} (J(\boldsymbol{\theta}^*) - J\boldsymbol{\theta}^n_{\mathrm{UCB}}) \leq \mathbb{E}_{\boldsymbol{s}_0}\left[\beta_N \sqrt{N}\sqrt{c_3 + c_4 M_N \log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N))}\right], \tag{55}$$

where

$$c_3 = 2c_1 \max_{\boldsymbol{x}\in\mathcal{X}} \sigma^2_{V,0}(\boldsymbol{x}), \tag{56}$$

$$c_4 = \frac{3}{2(1-\gamma)} c_2, \tag{57}$$

since $\gamma^{M_n} - \gamma^{M_n+1} \leq 1$. We can upper bound the argument of the expectation by taking the maximum over the data set in $\mathcal{Z}^{\sum_{n=1}^{N} M_n}$ with $\mathcal{Z} = \mathcal{X} \times \mathcal{S}$, which results in

$$\sum_{n=1}^{N} (J(\boldsymbol{\theta}^*) - J\boldsymbol{\theta}^n_{\mathrm{UCB}}) \leq \mathbb{E}_{\boldsymbol{s}_0}\left[\max_{\boldsymbol{z}_n\in\mathcal{Z}, n=1,\ldots,N} \beta_N \sqrt{N}\sqrt{c_3 + c_4 M_N \log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N))}\right] \tag{58}$$

$$\leq \max_{\boldsymbol{z}_n\in\mathcal{Z}, n=1,\ldots,N} \beta_N \sqrt{N} \sqrt{c_3 + c_4 M_N \Gamma_{k_\Delta}\left(\sum_{n=1}^{N} M_n\right)} \tag{59}$$

where we employ the maximum information gain Γ_{k_Δ} for the kernel k_Δ as defined in (7). Since the maximization over the data renders the argument of the expectation independent of the initial states, we drop the expectation. However, we keep the maximization over the data due to the non-explicitly dependency of β_N on the data as defined in (30). We further simplify this bound by noting that $\Gamma_{k_\Delta}(\sum_{n=1}^N M_n) \geq \Gamma_{k_\Delta}(0) \geq \log(1 + \sigma^{-2} \max_{\boldsymbol{z}\in\mathcal{Z}} \sigma_{\Delta,0}(\boldsymbol{z}))$ for $N \geq 1$ due to Lemma 5, such that we obtain

$$\sqrt{c_3 + c_4 M_N \Gamma_{k_\Delta}\left(\sum_{n=1}^N M_n\right)} \leq c_5\sqrt{M_N \Gamma_{k_\Delta}\left(N M_N\right)} \tag{60}$$

where

$$c_5 = \sqrt{\frac{c_3}{\log(1+\sigma^{-2}\max_{\boldsymbol{z}\in\mathcal{Z}}\sigma_{\Delta,0}(\boldsymbol{z}))} + c_4}. \tag{61}$$

For β_N, we can similarly proceed to bound its maximum over possible data sets by

$$\max_{\boldsymbol{z}_n\in\mathcal{Z}, n=1,\ldots,N} \beta_N \leq 2\sqrt{d_s L_k} B\lambda \sqrt{\Gamma_{k_\Delta}\left(N M_N\right) - 2\log\left(\frac{6\delta}{\pi^2 N^2}\right)} + \sigma B \tag{62}$$

using Lemma 5. Cauchy-Schwarz inequality together with $\Gamma_{k_\Delta}(\sum_{n=1}^N M_n) \geq \log(1 + \sigma^{-2}\max_{\boldsymbol{z}\in\mathcal{Z}} \sigma_{\Delta,0}(\boldsymbol{z}))$ for $N \geq 1$ results in

$$\begin{aligned}\max_{\boldsymbol{z}_n\in\mathcal{Z}, n=1,\ldots,N} \beta_N \leq & 2\sqrt{d_s L_k} B\lambda \Bigg(\left(2 + \frac{2\sigma^2 B^2}{\log(1+\sigma^{-2}\max_{\boldsymbol{z}\in\mathcal{Z}}\sigma_{\Delta,0}(\boldsymbol{z}))}\right)\Gamma_{k_\Delta}\left(N M_N\right) \\ & + 4\log\left(\frac{\pi^2 N^2}{6\delta}\right)\Bigg)^{\frac{1}{2}}.\end{aligned} \tag{63}$$

Due to $4\log\left(\frac{\pi^2 N^2}{6\delta}\right) = 4\log\left(\frac{\pi^2}{6\delta}\right) + 8\log(N)$ and the implication of $N \geq 2$ by the assumption of $N > 1$ for the integer N, we have $4\log\left(\frac{\pi^2}{6\delta}\right) + 8\log(N) \leq \left(4\log\left(\frac{\pi^2}{6\delta}\right)/\log(2) + 8\right)\log(N)$. Thus, we can further simplify the bound to

$$\max_{\boldsymbol{z}_n\in\mathcal{Z}, n=1,\ldots,N} \beta_N \leq c_6\sqrt{\Gamma_{k_\Delta}\left(N M_N\right) + \log(N)} \tag{64}$$

with

$$c_6 = 2\sqrt{d_s L_k} B\lambda \sqrt{\max\left\{\left(2 + \frac{2\sigma^2 B^2}{\log(1+\sigma^{-2}\max_{\boldsymbol{z}\in\mathcal{Z}}\sigma_{\Delta,0}(\boldsymbol{z}))}\right), \frac{4\log\left(\frac{\pi^2}{6\delta}\right)}{\log(2)} + 8\right\}}. \tag{65}$$

Finally, substituting (60) and (64) into (59), we obtain

$$\sum_{n=1}^N (J(\boldsymbol{\theta}^*) - J\boldsymbol{\theta}^n_{\mathrm{UCB}}) \leq c_5 c_6 \sqrt{N M_N \Gamma_{k_\Delta}\left(N M_N\right)\left(\Gamma_{k_\Delta}\left(N M_N\right) + \log(N)\right)} \tag{66}$$

□

C Bounding the Information Gain

Lemma 7. *Define* $\Xi = \mathrm{blkdiag}(\Xi_1, \ldots, \Xi_N)$*, where* $\Xi_n \in \mathbb{R}^{M_n\times(M_n+1)}$ *is given by*

$$\boldsymbol{\Xi}_n = \begin{bmatrix} 1 & -\gamma & 0 & \ldots & 0 \\ 0 & 1 & -\gamma & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & -\gamma \end{bmatrix}. \tag{67}$$

Moreover, let $\tilde{\boldsymbol{X}}_N$ *denote the matrix of all* $\boldsymbol{x}_m^n = [\boldsymbol{s}_m^n; \boldsymbol{\theta}^n]$ $m = 1, \ldots, M_n$*,* $n = 1, \ldots, N$*, arranged in the order they occur in the trajectories. Then, we have the identity*

$$\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)) = \boldsymbol{\Xi} K_V(\tilde{\boldsymbol{X}}_N)\boldsymbol{\Xi}^T. \tag{68}$$

Proof. To proof this identity, observe that we can express the temporal difference kernel k_Δ as the vector-matrix-vector product

$$k_\Delta(z_i, z_j) = \begin{bmatrix}1 & -\gamma\end{bmatrix} \begin{bmatrix} k_Q(x_i, x_j) & k_Q(x_i, [s'_j, \pi_{\theta_j}(s'_j), \theta_j]) \\ k_Q([s'_i, \pi_{\theta_i}(s'_i), \theta_i], x_j) & k_Q([s'_i, \pi_{\theta_i}(s'_i), \theta_i], [s'_j, \pi_{\theta_j}(s'_j), \theta_j]) \end{bmatrix} \begin{bmatrix} 1 \\ -\gamma \end{bmatrix}.$$

Considering the matrix $\tilde{\boldsymbol{X}}_1$ consisting of a single trajectory, this immediately implies that we have

$$K_\Delta(\boldsymbol{Z}_1, \boldsymbol{Z}_1) = \Xi K(\tilde{\boldsymbol{X}}_1, \tilde{\boldsymbol{X}}_1)\Xi^T. \tag{69}$$

We do not have a coupling via a γ-term between trajectories, such that we obtain a block-diagonal structure for Ξ when considering multiple trajectories resulting in (69). □

Lemma 8. *A block-diagonal matrix* Ξ *with blocks* (67) *satisfies*

$$\|\Xi\|^2 \leq 1 + \gamma^2, \tag{70}$$

where $\|\Xi\|$ *is the spectral norm of* Ξ*.*

Proof. The spectral norm of a matrix corresponds to its largest singular value, which in turn is the root of the maximum eigenvalue of $\Xi\Xi^T$. Thus, we have

$$\|\Xi\|^2 = \lambda_{\max}(\Xi\Xi^T). \tag{71}$$

Since Ξ is block-diagonal, the matrix $\Xi\Xi^T$ is also block-diagonal, i.e. $\Xi\Xi^T = \mathrm{blkdiag}(\Xi_1\Xi_1^T, \ldots, \Xi_N\Xi_N^T)$. Due to the block-diagonal structure of Ξ, its eigenvalues are given by the eigenvalues of its blocks such that

$$\|\Xi\|^2 = \max_{n=1,\ldots,N} \lambda_{\max}(\Xi_n\Xi_n^T). \tag{72}$$

Moreover, for each of its blocks, we can compute the matrix $\Xi_n\Xi_n^T$ in closed-form yielding

$$\Xi_n\Xi_n^T = \begin{bmatrix} 1+\gamma^2 & -\gamma & 0 & \ldots & 0 \\ -\gamma & 1+\gamma^2 & -\gamma & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1+\gamma^2 \end{bmatrix}. \tag{73}$$

Due to [42, Theorem 2.2], the maximum eigenvalue of such a tri-diagonal Toeplitz matrix is bounded by

$$\begin{aligned} \lambda_{\max}(\Xi_n\Xi_n^T) &= 1 + \gamma^2 - 2\sqrt{\gamma^2}\cos\left(\frac{M_n\pi}{M_n+1}\right), \qquad (74) \\ &\leq 1 + \gamma^2 + 2\gamma, \qquad (75) \\ &\leq (1+\gamma)^2 \qquad (76) \end{aligned}$$

which concludes the proof. □

Lemma 9. *Let* $\tilde{\boldsymbol{X}}_N$ *denote the matrix of all* $\boldsymbol{x}_m^n = [\boldsymbol{s}_m^n; \boldsymbol{\theta}^n]$ $m = 1, \ldots, M_n$*,* $n = 1, \ldots, N$*. Then, it holds that*

$$\log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)) \leq \log\det\left(\boldsymbol{I} + \frac{(1+\gamma)^2}{\sigma^2}\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N)\right). \tag{77}$$

Proof. Due to Lemma 7, we have the identity

$$\log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)) = \log\det(\boldsymbol{I} + \sigma^{-2}\Xi\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N)\Xi^T). \tag{78}$$

The log-determinant of a matrix corresponds to sum of the logarithm of its eigenvalues. Moreover, the eigenvalues $\lambda_i(\boldsymbol{I} + \boldsymbol{A})$ for an arbitrary quadratic matrix $\boldsymbol{A}$ are given by $1 + \lambda_i(\boldsymbol{A})$. Therefore, we obtain the identity

$$\log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)) = \sum_{i=1}^{NM_N} \log(1 + \sigma^{-2}\lambda_i(\Xi\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N)\Xi^T)). \tag{79}$$

The eigenvalues of the matrix $\Xi\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N)\Xi^T$ correspond to its singular values since it is positive definite. Similarly, the eigenvalues of $\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N)$ correspond to its eigenvalues as it is also positive definite. Hence, the min-max characterization of singular values guarantees

$$\lambda_i(\Xi\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N)\Xi^T) \leq \|\Xi\|^2\lambda_i(\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N)). \tag{80}$$

Substituting this bound into (79), we obtain

$$\log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)) = \sum_{i=1}^{NM_N} \log(1 + \sigma^{-2}\|\Xi\|^2\lambda_i(\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N))). \tag{81}$$

Finally, we employ Lemma 8, which results in

$$\log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_\Delta(\boldsymbol{Z}_N)) \leq \sum_{i=1}^{NM_N} \log(1 + \frac{(1+\gamma)^2}{\sigma^2}\lambda_i(\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N))), \tag{82}$$

$$\leq \log\det\left(\boldsymbol{I} + \frac{(1+\gamma)^2}{\sigma^2}\boldsymbol{K}_V(\tilde{\boldsymbol{X}}_N)\right), \tag{83}$$

where the second line follows from the equivalence of the log-determinant of a matrix and the sum over the logarithm of its eigenvalues. □

Theorem 2. *The maximum information gain for the temporal difference kernel k_Δ is bounded by the information gain of its base kernel, i.e.,*

$$\Gamma_{k_\Delta}(NM_N) \in \mathcal{O}(\Gamma_{k_V}(NM_N)) \tag{84}$$

for a non-decreasing sequence M_n, $n = 1, \ldots, N$.

Proof. We consider the scaling factor $1+\gamma^2$ on the right side of (77) as part of the variance, i.e., assume a noise variance $\tilde{\sigma}^2 = \frac{\sigma^2}{1+\gamma^2}$. Then, the right side corresponds to $\Gamma_{k_V}(N(M_N+1))$. As the scaling factor is constant, it does not influence the asymptotic behavior for growing N, which yields (84). □