Gaussian Process Q-Learning for Finite-Horizon Markov Decision Process

Anonymous authors
Paper under double-blind review

Keywords: Finite Horizon Markov Decision Process, Gaussian Process, Q-learning

Summary

Many real-world control and optimization problems require making decisions over a finite horizon to maximize performance. This paper proposes a reinforcement learning framework that approximately solves the finite-horizon Markov Decision Process (MDPs) by combining Gaussian Processes (GP) with Q-learning. The method addresses two key challenges: the tractability of exact dynamic programming in continuous state-control spaces, and the need for sample-efficient state-action value function approximation in systems where data collection is expensive. Using GPs and backward induction, we construct state-action value function approximations that enable efficient policy learning with limited data. To handle the computational burden of GPs as data accumulates across iterations, a subset selection mechanism is introduced which uses M-determinantal point processes that draw diverse, high-performing subsets. Theoretical analysis establishes probabilistic uniform error bounds on the convergence of the GP posterior mean to the optimal state-action value function for convex MDPs with deterministic dynamics. Empirical case studies are explored through a linear quadratic regulator problem and online optimization of a non-isothermal semi-batch reactor. Improved learning efficiency is shown relative to the use of proximal policy optimization with a neural network policy and state-action value function approximation.

Contribution(s)

1. The paper presents a framework for learning Gaussian process state-action value function approximations using Q-learning for deterministic finite horizon Markov Decision Processes.
   Context: Prior work has explored the use of Gaussian process models within the infinite horizon context (Engel et al., 2005; Grande et al., 2014; Chowdhary et al., 2014).
2. A subset selection strategy is proposed to ensure the online computational tractability of control inference using M-determinantal point processes to build a GP approximation of the state-action value function, that balances global coverage with local accurate modeling in highly performing regions of the state-control space (Kulesza et al., 2012; Moss et al., 2023)
   Context: Previous works take the approach of building variational approximations globally to the exact GP state-action value function approximation (Grande et al., 2014).
3. We provide probabilistic error bounds and asymptotic convergence rates for the error in modeling the optimal state-action value function for convex MDPs with deterministic dynamics.
   Context: We build on previous work on probabilistic error bounds derived in Lederer et al. (2021b).

Gaussian Process Q-Learning for Finite-Horizon Markov Decision Process

Anonymous authors
Paper under double-blind review

Abstract

Many real-world control and optimization problems require making decisions over a finite time horizon to maximize performance. This paper proposes a reinforcement learning framework that approximately solves the finite-horizon Markov Decision Process (MDP) by combining Gaussian Processes (GPs) with Q-learning. The method addresses two key challenges: the tractability of exact dynamic programming in continuous state-control spaces, and the need for sample-efficient state-action value function approximation in systems where data collection is expensive. Using GPs and backward induction, we construct state-action value function approximations that enable efficient policy learning with limited data. To handle the computational burden of GPs as data accumulate across iterations, we propose a subset selection mechanism that uses M-determinantal point processes to draw diverse, high-performing subsets. Theoretical analysis establishes probabilistic uniform error bounds on the convergence of the GP posterior mean to the optimal state-action value function for convex MDPs with deterministic dynamics. The proposed method is evaluated on a linear quadratic regulator problem and online optimization of a non-isothermal semi-batch reactor. Improved learning efficiency is shown relative to the use of proximal policy optimization with a neural network policy and state-action value function approximation.

1 Introduction

Sequential decision-making problems with finite time horizons appear across numerous domains. These problems are often naturally modeled as finite-horizon Markov Decision Processes (MDPs), where decisions must optimize the total cost incurred over a predetermined time window. Fed-batch process control and online optimization is a prominent example within the process systems engineering (PSE) community (Mesbah, 2016; Bradford et al., 2020), where optimal policy identification is inherently challenging due to difficulties in system model identification and uncertainty propagation over the decision horizon. While optimal solutions to these problems can theoretically be obtained through dynamic programming (DP), three significant challenges emerge in practice: (i) the intractability of exact DP in continuous state-action spaces, (ii) the lack of an exact descriptive process model, and (iii) the need for sample efficiency when data collection is expensive.

This paper introduces a reinforcement learning framework that addresses these challenges by combining the statistical modeling power of Gaussian Processes (GPs) with Q-learning. Previous works have predominantly explored the use of GPs to approximate state-space models (Kuss & Rasmussen, 2003; Deisenroth et al., 2015; Bradford et al., 2020; Mowbray et al., 2022), with relatively few taking a purely model-free approach. For example, Savage et al. (2021) exploited GP models for state-action value learning. However, the paper provides no mechanism for updating the underlying dataset, for example through a Monte Carlo or Q-learning estimator and therefore is not guaranteed to identify the optimal policy. The GP temporal difference learning (GPTD) algorithm detailed in Engel et al. (2005) is tailored for prediction problems by explicitly modeling rewards and exploiting the structure of the Bellman equation, to identify the value function of a stationary policy that has generated the modeled data. The algorithm can be extended for on-policy control through the use of SARSA. Nevertheless, the authors do not exploit the uncertainty of the state-action value function posterior process to balance exploration and exploitation; and, as we argue subsequently, off-policy updates are preferable. Chowdhary et al. (2014) have explored the development of GP-based Q-learning algorithms; however convergence is dependent upon non-trivial selection of a regularization parameter. Grande et al. (2014) provide a GPQ learning algorithm with an update rule that assumes the observed rollout estimate and the approximation belong to a joint Gaussian distribution. This assumption likely only holds for a subset of MDPs.

Unlike traditional state-action value function approximation methods that rely on neural networks (Azizzadenesheli & Anandkumar, 2019) or linear models, GPs offer significant advantages: they provide epistemic and aleatoric uncertainty quantification that can be exploited for decision making through the use of bandit strategies, can incorporate prior domain knowledge through kernel selection, and inherently balance model complexity as data are collected.

The contributions of this work are as follows. We pose a framework for learning optimal control using Q-learning and GPs in finite-horizon MDPs with deterministic dynamics and justify the composition of the two. The computational challenges associated with the use of GPs are handled through implementation of a principled subset selection mechanism using M-determinantal point processes (M-DPPs) (Kulesza & Taskar, 2012; Moss et al., 2023). This mechanism enables efficient scaling of control inference with increasing dataset size, despite the cubic complexity of exact GP inference. We provide theoretical analysis showing probabilistic uniform error bounds on the convergence of the GP posterior mean to the optimal state-action value functions, specifically examining how approximation errors propagate through finite horizons in convex MDPs. Empirical validation on both a linear quadratic regulator (LQR) system and a non-isothermal semi-batch chemical reactor optimization problem demonstrate the algorithm's sample efficiency. This is a significant advantage in real-world processes where experiments involve substantial costs and time investment.

In summary, we develop a GP-based Q-learning framework for finite-horizon MDPs, implement a practical approach using M-DPPs for strategic subset selection, and establish theoretical guarantees on convergence within the domain of convex finite-horizon MDPs. The rest of this paper is structured as follows: Section 2 presents preliminaries, Section 3 the GPQL algorithm, Section 4 the case studies, conclusions in Section 5 and analysis in Appendix A.2

2 Preliminaries

2.1 Finite-horizon Markov decision process

Consider a finite-horizon Markov Decision Process (MDP) which is defined by the 5-tuple $< \mathcal{X}, \mathcal{U}, \mathcal{P}, \Phi, \mathcal{T} >$. Specifically, the states are described by a compact state set, $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{n_x}$, with the controls restricted to a compact set, $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{n_u}$. The underlying decision process adheres to discrete-time state transitions described by a conditional probability density function, $\mathcal{P} : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to [0, \infty)$ over a time horizon, $\mathcal{T} = \{0, \dots, T\}$. For convenience, in the following we denote the state-control pair, $\mathbf{z} = [\mathbf{x}, \mathbf{u}]^\intercal$, and corresponding set, $\mathcal{Z} = \mathcal{X} \times \mathcal{U}$. We restrict the presentation to consider deterministic state-transitions, such that one may define

$$\mathcal{P}(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t) = \prod_{i=1}^{n_x} \delta(x_{i,t+1} - f_i(\mathbf{x}_t, \mathbf{u}_t)),$$

where $\delta(\cdot)$ indicates the dirac-delta function. $x_{i,t} \in \mathbb{R}$ the i^{th} state component, and $\mathbf{f}(\mathbf{x}, \mathbf{u}) = [f_1(\mathbf{x}, \mathbf{u}), \dots, f_{n_x}(\mathbf{x}, \mathbf{u})]^\intercal$ a discrete-time dynamic model of the system,

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t). \tag{1}$$

The objective of decision making is to select controls, according to a non-stationary state-feedback policy, $\pi(\cdot) = (\pi_0(\cdot), \dots, \pi_{T-1}(\cdot))$, with $U_t \sim \pi_t(\mathbf{u}_t \mid \mathbf{x}_t)$, to minimize the expected sum of stage-costs allocated by a cost function, $\Phi(\mathbf{x}, \mathbf{u})$, incurred over a finite discrete-time horizon,

$$\begin{aligned} &\min_{\pi} \mathbb{E}_{\pi}\left[Q_t^{\pi}(X_0, U_0) \mid X_0 = \mathbf{x}_0\right] \\ &Q_t^{\pi}(\mathbf{x}, \mathbf{u}) \coloneqq \Phi(\mathbf{x}, \mathbf{u}) + \mathbb{E}_{\pi}\left[\sum_{k=t+1}^{T-1} \Phi(X_k, U_k) \middle| U_k \sim \pi_k(\mathbf{u}_k \mid \mathbf{x}_k)\right] . \end{aligned} \tag{2}$$

Note that although the initial state, $\mathbf{x}_0$, is assumed to be known with certainty, in general the state itself is uncertain due to the definition of the policy as a conditional probability density function.

According to the principle of optimality, however, the optimal policy is a deterministic function of state and acts to greedily minimize the state-action value function at a given time within the horizon,

$$\pi^*(\mathbf{x}_t) \in \arg\min_{\pi} \; Q_t^{\pi}(\mathbf{x}_t, \pi_t(\mathbf{x}_t)) . \tag{3}$$

In principle, the optimal policy can be identified through dynamic programming (DP). However, DP becomes intractable in large or continuous state-control spaces and is reliant on a known model of the system, which is unavailable in general. This directs attention to the use of model-free approximate methods, such as Q-learning, which, in the tabular sense, aim to approximate the state-action values associated with state-control pairs in a given dataset,

$$\mathcal{D} = \{\mathcal{D}_t\}_{t=0:T-1}\,, \quad \mathcal{D}_t = \left\{\left(\mathbf{x}_t^{(\mathrm{i})}, \mathbf{u}_t^{(\mathrm{i})}, \mathbf{x}_{t+1}^{(\mathrm{i})}, \Phi(\mathbf{x}_t^{(\mathrm{i})}, \mathbf{u}_t^{(\mathrm{i})})\right)\right\}_{i=1:N} .$$

As the cardinality of dataset, N, increases, the approach provided by maintaining fixed point estimates of the state-action value function faces an explosion in memory cost. This challenge is typically handled through the construction of function approximators, e.g., deep Q-learning.

2.2 Gaussian process

GPs are probabilistic models that describe the relationship between a finite number of evaluations of a multivariate, scalar-valued function via a joint multivariate Gaussian distribution. Consider the input locations, $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$, corresponding noiseless function evaluations, $Y = \{f(\mathbf{z}_1), \dots, f(\mathbf{z}_N)\}$, test location, $\mathbf{z}_*$, and function value, y_*. The zero-mean GP prior asserts,

$$\begin{bmatrix} Y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} K(Z,Z) & \mathbf{k}(Z,\mathbf{z}_*) \\ \mathbf{k}(Z,\mathbf{z}_*)^{\intercal} & k(\mathbf{z}_*,\mathbf{z}_*) \end{bmatrix}\right) \tag{4}$$

with $k(\mathbf{z}, \mathbf{z}') \in \mathbb{R}$ denoting the kernel function, $\mathbf{k}(Z, \mathbf{z}_*) \in \mathbb{R}^N$ the covariance of the function values between the input locations and test point, $k(\mathbf{z}_*, \mathbf{z}_*) \in \mathbb{R}$ the variance of the function value at the test point, and $K(Z, Z)$ denoting the gram matrix representative of the covariance of the function values evaluated at the input locations, Z. Bayesian inference gives the predictive posterior distribution:

$$\begin{aligned} \mu(\mathbf{z}_*) &= \mathbf{k}(Z,\mathbf{z}_*)^{\intercal} K(Z,Z)^{-1} Y \\ \mathrm{k}(\mathbf{z}_*,\mathbf{z}_*) &= k(\mathbf{z}_*,\mathbf{z}_*) - \mathbf{k}(Z,\mathbf{z}_*)^{\intercal} K(Z,Z)^{-1} \mathbf{k}(Z,\mathbf{z}_*)\,, \end{aligned} \tag{5}$$

with $y_* \sim \mathcal{N}\left(\mu(\mathbf{z}_*), \mathrm{k}(\mathbf{z}_*, \mathbf{z}_*)\right)$ providing a description of the posterior at $\mathbf{z}^*$. One may also be interested in a general description of the posterior process,

$$f(\cdot) \mid Y \sim \mathcal{GP}\left(\mu(\cdot), \mathrm{k}(\cdot,\cdot)\right) , \tag{6}$$

which may be used to approximate the state-action value function using the available data. A policy may then be defined to exploit the posterior process for decision-making, generalizing to continuous state-control spaces. Such a policy may be defined by greedily exploiting functions drawn from the posterior, which involves solving minimization problems. This motivates discussion regarding properties of the optimal solution map and value function.

2.3 Optimal value functions and solution maps of nonlinear programs

Consider the optimal (possibly point-to-set) solution map, $\mathbf{u}^* \in G^*(\mathbf{x})$ and optimal value function, $g^*(\mathbf{x})$, of the following parametric nonlinear program with constant constraint map (Fiacco & Ishizuka, 1990),

$$\begin{aligned} G^*(\mathbf{x}) &:= \arg\min_{\mathbf{u}\in\mathcal{U}} \ g(\mathbf{x},\mathbf{u}) \\ g^*(\mathbf{x}) &:= \min_{\mathbf{u}\in\mathcal{U}} \ g(\mathbf{x},\mathbf{u})\,, \end{aligned} \tag{7}$$

with $g(\mathbf{x},\mathbf{u})$ indicating a multivariate, scalar valued objective function.

Assumption 1 *The nonlinear objective function is continuous in both* $\mathbf{x}$ *and* $\mathbf{u}$*, with* $(\mathbf{x},\mathbf{u}) \in \mathcal{Z}$*.*

Property 1 *Let Assumption 1 hold. The optimal value function,* $g^*(\mathbf{x})$*, is continuous (Fiacco & Ishizuka, 1990; Aubin & Frankowska, 1999), and the optimal solution map,* $G^*(\mathbf{x})$*, is upper semi-continuous on* $\mathbf{x} \in \mathcal{X}$ *(Sundaram, 1996).*

Strict convexity assumptions are typically required for the solution map to be single-valued and continuous (Fiacco & Ishizuka, 1990; Aubin & Frankowska, 1999).

3 Methodology

In the following section, we present a methodology that leverages GPs to approximate the state-action value function. This directs the presentation of the following problem statement.

3.1 Problem statement

Consider the finite-horizon MDP introduced in Section 2.1. Additionally, for simplicity in subsequent analysis, the following assumptions are imposed on the dynamics and cost function to ensure that the state-action value function could be well approximated by a continuous function.

Assumption 2 (Continuous Dynamics) *The underlying state transition probability density function,* $\mathcal{P}(\cdot)$*, can be parameterized by* $\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t, \mathbf{u}_t)$*, where* $\mathbf{f}_t : \mathcal{Z} \to \mathcal{X}$ is a continuous function, $\forall t \in \{0, \dots, T-1\}$.

Assumption 3 (Policy Continuity) *The policy,* $\pi(\mathbf{u} \mid \mathbf{x})$ *is a conditional probability density function continuous with respect to* $\mathbf{x}$ *and with support provided by the compact set,* $\mathcal{X}$*.*

Property 2 (State-Action Value Function Continuity) *Assume that the cost function,* $\Phi : \mathcal{Z} \to \mathbb{R}$ *is continuous over the compact set* $(\mathbf{x},\mathbf{u}) \in \mathcal{Z}$*. Together with Assumptions 2–3, this implies the state-action value function induced under a policy,* $Q_t^\pi : \mathcal{Z} \to \mathbb{R}_+$ *is continuous* $\forall t \in \{0, \dots, T-1\}$*.*

The reasoning behind Property 2 follows (i) the expectation taken over an input continuous probability density function yields the expectation itself input continuous, and (ii) the composition of continuous functions yields a continuous function.

Proposition 1 *Let Property 2 hold. The value function associated with the nonlinear program,*

$$\min_{\mathbf{u}\in\mathcal{U}} \ Q_t^\pi(\mathbf{x},\mathbf{u})\,,$$

satisfies Property 1.

Theorem 1 *Let Proposition 1 hold, then the optimal state-action value function,* $Q_t^{\pi^*}(\mathbf{x},\mathbf{u})$ *is a continuous function defined on* $(\mathbf{x},\mathbf{u}) \in \mathcal{Z}$*,* $\forall t \in \{0, \dots, T-1\}$*. This can be proven by backward induction starting from the terminal cost function.*

.

3.2 Gaussian process Q-Learning (GPQL)

Having formalized the problem setting, we now direct our attention to the contribution of this work. Namely, we introduce Gaussian process Q-Learning (GPQL), which is well suited to solving the problems described, and adheres to the general framework for policy iteration.

3.2.1 Gaussian process state-action value function approximation

We propose to approximate the state-action value function, $Q_t^\pi(\cdot)$ using GP models and consider the construction of independent GP models for each time-step. Specifically, we assume a dataset, $\mathcal{D}_t\ \forall t$, which provides some discretization of the state-control space, $\mathbf{z} \in \mathcal{Z}$, from which state-action values may be estimated via fixed-point estimates, $\mathbf{Q}_t^\pi = \{Q_t^\pi(\mathbf{z})\ \forall \mathbf{z} \in \mathcal{D}_t\}$, as in the tabular setting. A joint prior distribution defines the predictive model of the state-action value, $Q_{t,*}^\pi$, at a new state-control pair, $\mathbf{z}_* \in \mathcal{Z}$,

$$\begin{bmatrix} \mathbf{Q}_t^\pi \\ Q_{t,*}^\pi \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} K_t(Z,Z) & \mathbf{k}_t(Z,\mathbf{z}_*) \\ \mathbf{k}_t(Z,\mathbf{z}_*)^\intercal & k_t(\mathbf{z}_*,\mathbf{z}_*) \end{bmatrix} \right) . \tag{8}$$

Through conditioning the posterior process may be obtained, $Q_t^\pi(\cdot) \mid \mathbf{Q}_t^\pi \sim \mathcal{GP}\left(\mu_t^\pi(\cdot), \mathrm{k}_t^\pi(\cdot,\cdot)\right)$, as defined in Section 2.2. In principle, given knowledge of the underlying problem structure, one can select a kernel appropriately, such that the state-action value function lies within the class of functions well represented by the GP. Considerations for selection are outlined later in Section 3.2.3.

It is worth highlighting that the complexity of GP model inference scales cubically with the number of datapoints, N. In practice, we use the M-determinantal point process (M-DPP) to select a subset of the state-action value point estimates to build an approximation. A subset of $M \leq N$ points, Z_M, are sampled from $\mathcal{D}_t$, according to their probability under the M-DPP,

$$\begin{aligned} \mathbb{P}(Z_M \subseteq Z) &\propto |L_{Z_M}| \\ |L_{Z_M}| &= |K(Z_M, Z_M)| \cdot \prod_{\mathbf{z} \in Z_M} q(\mathbf{z})^2 , \end{aligned} \tag{9}$$

which is proportional to the trace of the Gram matrix weighted by the quality of the points as evaluated by $q(\mathbf{z})$, $\forall \mathbf{z} \in Z_M$. This is a strategy that was reported in Kulesza et al. (2012), and has since been utilized within the context of Bayesian optimization with sparse GPs (Moss et al., 2023). Intuitively, it selects points to balance performance under the quality function and coverage of the state-control space as evaluated by the kernel. Here, we utilize the strategy to simply build an exact GP estimator providing a local model in promising regions of the state-control space. The greedy method utilized in Chen et al. (2018); Moss et al. (2023) is implemented for convenience.

3.2.2 Thompson sampling for policy improvement

Within the course of approximate policy improvement, the GP model is exploited to generate a policy which balances exploration and exploitation. Specifically, a Thompson sampling (TS) policy, $\pi(\mathbf{u} \mid \mathbf{x})$, is deployed (Wilson et al., 2020). The policy is defined through minimization of state-action value functions, $Q_t^\pi(\cdot)$, sampled probabilistically from the GP posterior,

$$\pi_t(\mathbf{u} \mid \mathbf{x}) = \Pr\left(\mathbf{u} \in \arg\min_{\bar{\mathbf{u}} \in \mathcal{U}} Q_t^\pi(\mathbf{x}, \bar{\mathbf{u}}) \mid \mathbf{Q}_t^\pi \;\middle|\; Q_t^\pi(\cdot) \mid \mathbf{Q}_t^\pi \sim \mathcal{GP}\left(\mu_t^\pi(\cdot), \mathrm{k}_t^\pi(\cdot,\cdot)\right) \right) . \tag{10}$$

This enables one to exploit the GP-based state-action value function approximation to both explore and exploit the decision-space—effectively leveraging the quantification of epistemic uncertainty to generate data under a behavior policy, which may then be used to learn the target optimal policy.

Remark 1 *In principle, the TS behavior policy is upper semi-continuous in the state, and does not satisfy Assumption 3. However, this does not impact the state-action value function approximation problem posed as discussed in the subsequent section.*

3.2.3 Q-Learning with backward induction

Having defined means to both approximate the state-action value function using point-estimates and exploit it for decision-making, we now discuss updates of the point estimates, $\mathbf{Q}_t^\pi$. In any given rollout of the TS policy, one generates data that may be used to update the existing dataset,

$$\mathcal{D}_{\mathsf{n},t} = \left\{ \left(\mathbf{x}_t^{(\mathsf{n})}, \mathbf{u}_t^{(\mathsf{n})}, \Phi(\mathbf{x}_t^{(\mathsf{n})}, \mathbf{u}_t^{(\mathsf{n})}), \mathbf{x}_{t+1}^{(\mathsf{n})} \right) \right\} , \quad \mathcal{D}_t \leftarrow \mathcal{D}_t \cup \mathcal{D}_{\mathsf{n},t} \quad \forall t \, .$$

This new dataset may be used to update the GP posterior state-action value function approximations working from the final decision-step, given $\mathbf{Q}_{T-1}^\pi = \{\Phi(\mathbf{x}_{T-1}, \mathbf{u}_{T-1}),\ \forall (\mathbf{x}_{T-1}, \mathbf{u}_{T-1}) \in \mathcal{D}_{T-1}\}$, and propagating information backwards over the horizon via the following update,

$$\begin{aligned} Q_t^\pi(\mathbf{x}_t, \mathbf{u}_t) \leftarrow Q_t^\pi(\mathbf{x}_t, \mathbf{u}_t) + \alpha \left(\Phi(\mathbf{x}_t, \mathbf{u}_t) + \min_{\mathbf{u}_{t+1} \in \mathcal{U}} \mu_{t+1}^\pi(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) - Q_t^\pi(\mathbf{x}_t, \mathbf{u}_t) \right) \\ \forall \, (\mathbf{x}_t, \mathbf{u}_t, \Phi(\mathbf{x}_t, \mathbf{u}_t), \mathbf{x}_{t+1}) \in \mathcal{D}_t, \quad \forall t < T-1 \, , \end{aligned} \tag{11}$$

with $\alpha \in (0, 1]$ denoting a Robbins Munro step-size. This is a deterministic update rule based on the mean posterior GP approximation of the future cost-to-go. In principle, in the deterministic finite horizon setting one may keep $\alpha = 1$; however, we found Robbins-Munro step sizes empirically preferable. The updated state-action value point-estimates are then used to construct updated GPs.

The following analysis is provided as a comment on the considerations for approximation.

Assumption 4 *The cost function, $\Phi : \mathcal{Z} \to \mathbb{R}$ is a strictly convex function draw from $\Phi \sim \mathcal{GP}(0, k(\cdot, \cdot))$ with convex kernel, $k \in \mathcal{K}_{PD}$, on the compact set, $\mathcal{Z} \times \mathcal{Z}$.*

Theorem 2 *Let Assumption 2 and 4 hold and ensure that the state-action value point estimates, $\boldsymbol{Q}_{T-1}^\pi$, are updated through (11). For the general case of nonlinear dynamics, the point estimates do not lie in the same class of functions as the cost function in Assumption 4.*

Theorem 3 *Assume an initial dataset generated through a policy satisfying Assumption 3 from a finite horizon MDP satisfying Assumptions 2 and 4. Provided the optimal future cost-to-go estimate from the posterior mean in (11) is the value function of a nonlinear program satisfying Property 1, the state-action value estimates are described by a continuous function for all timesteps.*

Theorems 2 and 3 motivate the use of the off-policy Q-learning update[1]. The implication is that, regardless of the behavior policy, the state-action value estimates yielded by (11) are well approximated by a continuous function, although one not necessarily within the same class as the cost function. The proof of Theorem 2 follows trivially from the assumption of nonlinear dynamics. The proof of Theorem 3 is general beyond GP approximation and follows the reasoning from Section 2.3. We note that, if the initial step size is set $\alpha = 1$, then the assumption for the initial state-action value estimates to have been generated by a policy satisfying Assumption 3 may be relaxed. The practical implications of Theorem 2 are relatively minimal given the establishment of representer theorems (Williams & Rasmussen, 2006) and approximation capacity of universal kernels (Micchelli et al., 2006). In the case of affine dynamics and convex cost the following Corollary can be stated.

Corollary 1 *Assume time-varying or time-constant affine dynamics, and a cost function, $\Phi : \mathcal{Z} \to \mathbb{R}_+$, strictly convex on the compact space, $\mathcal{Z}$. Under the assumption of a strictly input convex kernel, positive definite Gram matrix, and restriction for all $\boldsymbol{Q}_{t+1}^\pi$ be strictly non-negative, then the optimal future cost-to-go is a convex function of the current state-control pair. Hence the iterates yielded by (11) are well approximated by a convex function.*

Corollary 1 is exemplified by the case of the LQR, where the state-action value function is a quadratic function of the state and control, as is the cost function, for all time indices in the horizon.

[1]Note that these results do not necessarily hold for on-policy updates such as SARSA if the rollout policy does not have continuity properties, which is often the case.

The restriction imposed on all $\mathbf{Q}_{t+1}^{\pi}$ to be non-negative can be enforced by simply bounding the iterates produced by (11). Algorithm 1 describes GPQL as proposed. In the Appendix A asymptotic analysis and probabilistic error bounds are provided for for the case of a convex MDP with deterministic dynamics, building upon Corollary 1.

Algorithm 1 Gaussian Process Q-Learning

```
1:  **Input:** Dynamics, $\mathcal{P}(\cdot)$ with compact state set $\mathcal{X}$, compact control set $\mathcal{U}$, a cost function, $\Phi(\cdot)$, and discrete
    time horizon, $t \in \{0, \dots, T-1\}$. A posterior Gaussian process state-action value function approximation
    $Q_t^{\pi}(\cdot) \mid \mathbf{Q}_t^{\pi} \sim \mathcal{GP}(\mu_t^{\pi}, \mathrm{k}_t^{\pi}(\cdot,\cdot)),\ \forall t$, estimated from initial data, $\mathbf{Q}_t^{\pi}$, and an initial policy $\pi_t$.
2:  **while** evaluation budget available **do**
3:      **Approximate Policy Evaluation:** Update $Q_t^{\pi}(\cdot) \mid \mathbf{Q}_t^{\pi} \sim \mathcal{GP}(\mu_t^{\pi}, \mathrm{k}_t^{\pi}(\cdot,\cdot)),\ \forall t$
4:      Rollout the policy, $\pi(\cdot)$ by sampling $\mathcal{P}(\cdot)$
5:      Collect $(\mathbf{x}_t, \mathbf{u}_t, \Phi(\mathbf{x}_t, \mathbf{u}_t), \mathbf{x}_{t+1})$ and store in $\mathcal{D}_t,\ \forall t$.
6:      **for** $t \in \{T-1, \dots, 0\}$ **do**
7:          Update the data $\mathbf{Q}_t^{\pi}$ according to (11).
8:          Update the posterior process $Q_t^{\pi}(\cdot) \mid \mathbf{Q}_t^{\pi} \sim \mathcal{GP}(\mu_t^{\pi}, \mathrm{k}_t^{\pi}(\cdot,\cdot))$
9:      **end for**
10:     **Approximate Policy Improvement:** Update the policy, $\pi$
11:     Define the Thompson Sampling policy, $\pi$, through the updated posterior processes via (10).
12: **end while**
13: **Return:** Optimized policy $\pi^*$ and function approximation $Q^*$.
```

4 Experiments

We evaluate GPQL computationally on two control tasks: an LQR problem, and a non-isothermal semi-batch reactor, which highlights the practical applicability of our method to real-world systems.

4.1 Linear Quadratic Regulator

The LQR problem provides an ideal benchmark for our approach as it satisfies Assumptions 2 and 7 with its affine dynamics, exhibits the convexity properties required by Properties 2 and 5, and exemplifies the conditions in Corollary 1 where the state-action value function belongs to the same function class as the cost function. We formulate an LQR problem using a double integrator system (position and velocity control) with dynamics and cost matrices defined in Appendix B.1 with time horizon $T = 6$. For this system, we employed a positive definite convex kernel to align with Corollary 1 and initialized with 5 episodes generated from a Sobol sequence on control trajectories. Figure 1 shows the final policy[2] that our GPQL algorithm identifies consistently approaches the oracle's performance after collection of 25 further batches[3]. We compare to a standard PPO (Schulman et al., 2017) implementation and found that it requires approximately 5000 batches to reach the same level of performance that GPQL achieves in a total of 30. Figure 2 illustrates the convergence by showing the state and control trajectories. The algorithm's learned policy (blue) closely aligns with the oracle's optimal trajectories (red) after training. This demonstrates that our method not only minimizes cost but also recovers the underlying optimal control structure of the LQR problem with relatively small datasets.

4.2 Semi-Batch Reactor

The second experiment involves a benchmark non-isothermal semi-batch reactor control problem (Bradford & Imsland, 2018; Bloor et al., 2024). The system expresses a series reaction mechanism:

[2]The final policy greedily exploits the posterior mean function.

[3]Here a batch refers to a rollout of the policy within the system over the discrete time horizon.

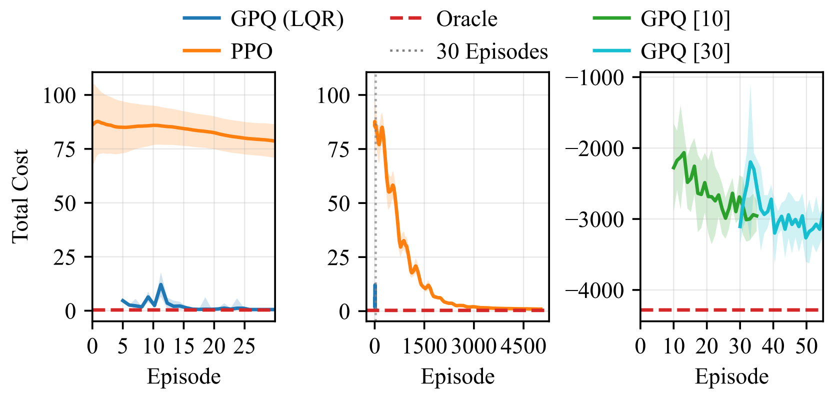

Figure 1: From left to right: Learning curve for the **LQR** case study for GPQL training. Learning curve for the **LQR** case study for the entire PPO training. Learning curve for the **Semi-batch Reactor** case study. GPQL shown in blue for the LQR case with initial dataset size of 5, and in blue, green and cyan for the Semi-batch Reactor case with initial dataset sizes of 10 and 30, respectively. PPO is orange and the oracle in dashed red. The shaded area shows a single standard deviation.

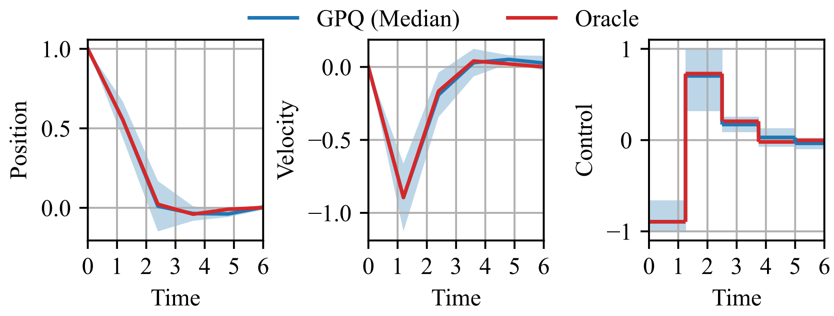

Figure 2: State and control trajectories for the LQR case study. Median states and controls from five repeats (blue) closely align with the oracle's optimal trajectories (red). Shaded areas represent one standard deviation.

$A \xrightarrow{k_1} B \xrightarrow{k_2} 3C$, where the first reaction is exothermic and the second is endothermic. The state vector $\mathbf{x} = [C_A, C_B, C_C, T, V]^\intercal$ tracks species concentrations, temperature, and reactor volume, while control inputs $\mathbf{u} = [T_c, F]^\intercal$ represent cooling jacket temperature and feed flow rate. The objective is to maximize the final amount of product C—the cost function is detailed in the supplementary.

Due to the nonlinearity of this system, we employed a Matérn-5/2 kernel for our GP models. We evaluated GPQL with different initial dataset sizes (10 and 30 batches). We allow the TS policy to collect a further 25 batches and compare the final policy (greedily exploiting the posterior mean function) against an oracle nonlinear model predictive controller (NMPC) with perfect system model. Figure 1 shows the produced learning curves. With 30 initial batches, the starting performance is notably higher than with 10 batches, demonstrating the value of a larger initial dataset. An initial performance dip occurs in early training episodes due to the exploration behavior of TS.

Figure 3 compares the state and control trajectories to those of the oracle. The learned policy closely tracks the oracle except for early states and controls. As the initial dataset size increases to 30 batches, GPQL approaches the performance of the oracle, confirming that our method provides significant advantages in scenarios where experimental data is limited and costly.

5 Conclusions and Future Work

This paper introduces GPQL, an approach for solving finite-horizon MDPs by using GPs to approximate state-action value functions. Our method addresses the challenges of continuous state-control spaces while maintaining computational tractability through strategic subset selection. We establish

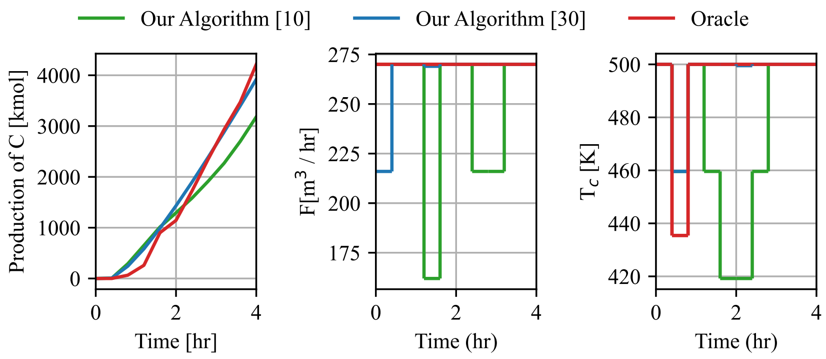

Figure 3: State and control trajectories for semi-batch reactor. The learned policy (blue) closely tracks the oracle (red) for key state variables and control inputs. Shaded areas represent one standard deviation.

theoretical guarantees on the continuity of state-action value functions and convergence properties, with particular analysis for convex MDPs. Our empirical evaluation on both LQR problems and a non-isothermal semi-batch reactor demonstrates that GPQL achieves near-optimal performance with limited data.

Future work includes extending our analysis to the case of uncertain dynamics. Additionally, investigating the application of the algorithm to MDPs with state chance constraints additionally imposed would enhance its practical utility for real-world control problems where state and control constraints are common.

A Analysis for Convex Markov Decision Process

A.1 Convex nonlinear programs

We start by extending the properties stated for the optimal solution maps and value functions for nonlinear programs.

Assumption 5 (Convex objective function) *The objective function $g(\mathbf{x}, \mathbf{u})$ is strictly convex in both arguments.*

Assumption 6 (Convex and compact constraint set) *Assume that $\mathcal{U} \subset \mathbb{R}^{n_u}$ denotes a non-empty compact and convex constraint set.*

Property 3 (Optimal solution mapping) *Let Assumptions 5 and 6 hold, then the optimal solution map, $G^*(\mathbf{x})$ is a single-valued mapping and continuous in $\mathbf{x}$ (Fiacco & Ishizuka, 1990).*

Property 4 (Value function convexity) *Let Assumptions 5 and 6 hold, then the value function, $g^*(\mathbf{x})$, is convex in $\mathbf{x}$ (Fiacco & Ishizuka, 1990).*

Remark 2 *With additional assumptions such as regularity conditions and higher order continuity of the constraint and objective functions, one can make stronger assertions regarding the differentiability of solution maps (Barratt, 2018).*

A.2 Convex Markov Decision Process

We begin by making the following assumptions on the problem statement.

Assumption 7 (Affine Dynamics and Convex Control Set) *Assume that the underlying state transition probability density function, $\mathcal{P}(\cdot)$, is parameterized by $\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t, \mathbf{u}_t)$, where $\boldsymbol{f}_t : \mathcal{Z} \rightarrow \mathcal{X}$ is an affine function, $\forall t \in \{0, \ldots, T-1\}$. The the control set, $\mathcal{U} \subset \mathbb{R}$, is compact and convex.*

Property 5 (Cost and State-Action Value Function Convexity) *Assume that the cost function, $\Phi : \mathcal{Z} \to \mathbb{R}_+$ is positive definite convex function on the compact set $(\mathbf{x}, \mathbf{u}) \in \mathcal{Z}$, with Lipschitz constant, L_Φ. Together with Assumptions 3 and 7, this implies the state-action value function induced under a policy, $Q_t^\pi : \mathcal{Z} \to \mathbb{R}_+$ is Lipschitz continuous with Lipschitz constant $L_{Q_t^\pi}, \forall t \in \mathcal{T}$ and a positive definite convex function in both arguments.*

The intuition behind Property 5 leverages (i) the additive structure of the state-action value function; (ii) the deterministic affine dynamics preserve the convexity of the future cost-to-go as a function of the current state-control pair; and (iii) that the expectation preserves convexity (Boyd, 2004).

Theorem 4 *Let Property 2 hold. The solution mapping and value function associated with the nonlinear program,*

$$\min_{\mathbf{u} \in \mathcal{U}} \; Q_t^\pi(\mathbf{x}_t, \mathbf{u}_t)\,,$$

satisfies Property 3 and 4, respectively.

An example of the setting posed here is the well-known LQR, where the dynamics are restricted to be affine, and the cost function a positive definite quadratic function of the state and control. It is well known that under these conditions the value function,

$$V_t^\pi(\mathbf{x}) = \mathbb{E}_\pi\left[Q_t^\pi(\mathbf{x}, U_t) \mid U_t \sim \pi_t\right]\,,$$

is convex quadratic in $\mathbf{x}$, and the state-action value function is quadratic is the state and control.

A.3 Gaussian process Q-learning through backward induction

We now formalize results on the approximation of state-action values for the convex MDP.

Assumption 8 *Assume the cost function, $\Phi : \mathcal{Z} \to \mathbb{R}_+$, is a strictly convex function drawn from a GP, $\mathcal{GP}(0, k(\cdot,\cdot))$, with strictly convex kernel function $k(\cdot,\cdot) \in \mathcal{K}_{PD}$ on the compact set, $\mathcal{Z} \times \mathcal{Z}$.*

Proposition 2 *Let the Gaussian posterior process, $\mathcal{GP}\left(\mu_t^\pi(\cdot), k_t^\pi(\cdot,\cdot)\right)$, be defined using a strictly input convex kernel function, $k(\cdot,\cdot)$, with a zero-mean prior. Let the initial observed function values, $\boldsymbol{Q}_\pi$ be non-negative and the Gram matrix be full rank and positive definite. The initial posterior process mean, $\mu_t^\pi(\mathbf{x}, \mathbf{u})$, is strictly convex on $(\mathbf{x}, \mathbf{u}) \in \mathcal{Z}$.*

Proposition 3 *Let Proposition 2 and Assumption 8 hold. The solution mapping, $\arg\min_{\mathbf{u} \in \mathcal{U}} \mu_t^\pi(\mathbf{x}, \mathbf{u})$, adheres to Property 3 and the optimal value function, $\min_{\mathbf{u} \in \mathcal{U}} \mu_t^\pi(\mathbf{x}, \mathbf{u})$, adheres to Property 4.*

Theorem 5 *Assume the problem statement provided in Appendix A.2 and Propositions 2 and 3. The iterative updates for point estimates of state-action values, $\boldsymbol{Q}_t^\pi$, generated by (11) satisfy Property 5 and belong to the same function class as the cost function.*

A.4 Analysis

In the following section, we analyze the asymptotic convergence of the mean of the posterior Gaussian process presented to the true optimal state-action value function and derive probabilistic uniform error bounds following the previous work of Lederer et al. (2021b). In this analysis we set $M = N$ within the M-DPP subset selection strategy (essentially switching it off) for construction of the GP models. Consider the problem statement provided in Appendix A.2, and let Assumption 8 hold. We proceed by firstly stating a probabilistic uniform error bound for the Lipschitz continuous and convex state-action value function at the final time step.

Theorem 6 *Assume N samples of the state-action value function at the final time step for control. A posterior process is obtained as,*

$$Q_{T-1}^\pi(\cdot) | \boldsymbol{Q}_{T-1}^\pi \sim \mathcal{GP}(\mu_{T-1}^\pi(\cdot), k_{T-1}^\pi(\cdot,\cdot)), \tag{12}$$

where the posterior mean $\mu^{\pi}_{T-1}(z)$ and variance $\sigma^2_{T-1}(z) = k^{\pi}_{T-1}(z,z)$ are continuous with Lipschitz constants $L_{\mu_{T-1}}$ and $L_{\sigma_{T-1}}$ on $\mathcal{Z}$. As proposed in Lederer et al. (2021a, Theorem 9), for a given $\delta_{T-1} \in (0,1)$ and grid-space constant, $\tau(N)$, used to discretize $\mathcal{Z}$ through the generation of $\boldsymbol{Q}^{\pi}_{T-1}$, one has,

$$\mathbb{P}\left(|Q^{\pi}_{T-1}(\mathbf{x},\mathbf{u}) - \mu^{\pi}_{T-1}(\mathbf{x},\mathbf{u})| \leq \sqrt{\beta_{T-1}(\tau)}\sigma_{T-1}(\mathbf{x},\mathbf{u}) + \gamma_{T-1}(\tau), \quad \forall(\mathbf{x},\mathbf{u}) \in \mathcal{Z}\right) \geq 1 - \delta_{T-1} \tag{13}$$

where,

$$\beta_{T-1}(\tau) = 2\log\left(\frac{M(\tau,\mathcal{Z})}{\delta_{T-1}}\right)$$
$$\gamma_{T-1}(\tau) = (L_{\mu_{T-1}} + L_{Q^{\pi}_{T-1}})\tau + \sqrt{\tau(N)\beta_{T-1}(\tau)L_{\sigma_{T-1}}}$$

with $M(\tau(N),\mathcal{Z})$ denoting the τ-covering number of $\mathcal{Z}$ and $L_{Q^{\pi}_{T-1}}$ denoting the Lipschitz constant of the state-action value function, $Q^{\pi}_{T-1}(\cdot)$.

By making additional assumptions on the data-generating policy, letting $\alpha = 1$ in (11), and following arguments made in Lederer et al. (2021b) we can establish an asymptotic convergence rate for the approximation of the state-action value function at the final time-step a decision is made.

Assumption 9 *The virtual grid constant $\tau(N)$ is chosen to decrease quadratically with the number of samples:*

$$\tau(N) \in \mathcal{O}(N^{-2})\,. \tag{14}$$

This assumption imposes the following requirement on the sampling policy $\pi_s(N)$: For any state $\mathbf{x} \in \mathcal{X}$, the policy must ensure that subsequent samples are taken with decreasing distance to the virtual grid points at a rate of at least $\mathcal{O}(N^{-2})$.

Theorem 7 (Asymptotic Convergence) *: Consider the posterior mean $\mu^{\pi}_{T-1}(\cdot)$ and the true function $Q_{T-1}(\cdot)$ defined on the compact set $\mathcal{Z}$. For any sequence of training data $\mathcal{D}_{T-1}$ as $N \to \infty$, if there exists a class $\mathcal{K}_\infty$ function $\alpha : \mathbb{R} \to \mathbb{R}^+$, where α is strictly increasing with $\alpha(0) = 0$ and $\lim_{r\to\infty} \alpha(r) = \infty$, with its convergence rate dependent on the choice of kernel such that*

$$\sigma_N(\mathbf{x},\mathbf{u}) \in \mathcal{O}\left(\frac{1}{\alpha(N)}\right) \subset \mathcal{O}\left(\frac{1}{\sqrt{\log(N)}}\right) \quad \forall(\mathbf{x},\mathbf{u}) \in \mathcal{Z}, \tag{15}$$

To analyze the convergence, the set $\mathcal{Z}$ is overapproximated by a hypercube with edge length θ. For this d-dimensional hypercube, the τ-covering number can be bounded by

$$M(\tau,\mathcal{Z}) \leq \left(\frac{\theta\sqrt{d}}{2\tau}\right)^d, \tag{16}$$

With the choice of virtual grid constant given in Assumption 9, the covering number grows polynomially with N:

$$M(\tau(N),\mathcal{Z}) \in \mathcal{O}(N^{2d}), \tag{17}$$

which leads to

$$\gamma(N) \in \mathcal{O}(N^{-1}), \tag{18}$$

Under these conditions, we can establish the following convergence rate:

$$\sup_{(\mathbf{x},\mathbf{u})\in\mathcal{Z}} \|\mu^{\pi}_{T-1}(\mathbf{x},\mathbf{u}) - Q^{\pi}_{T-1}(\mathbf{x},\mathbf{u})\| \in \mathcal{O}\left(\frac{\sqrt{\log(N)}}{\alpha(N)}\right) \quad a.s \tag{19}$$

With Lipchitz continuity of the posterior mean state-action value function approximation, the following derives a probabilistic error bound in approximation of the the optimal value function in the final decision step.

Proposition 4 (Final value function bound) *Let* $\mathbf{u}_g^* \in \arg\min_u g(\mathbf{x}, \mathbf{u})$ *and* $g^*(\mathbf{x}) = \min_{\mathbf{u}\in\mathcal{U}} g(\mathbf{x}, \mathbf{u})$ *then by the convexity of* $\Phi(\cdot)$ *and compactness of* $\mathcal{Z}$*, we have that*

$$|\mu^{\pi}_{T-1}(\mathbf{x}, \mathbf{u}^*_{Q_{T-1}}) - \mu^*_{T-1}(\mathbf{x})| \leq L_{\mu_{T-1}}|\mathbf{u}^*_{Q_{T-1}} - \mathbf{u}^*_{\mu_{T-1}}| \tag{20}$$

Proposition 5 (Final Optimal Value Function Bound) *Given the above, we can set a bound on the difference in the value function* $Q^*_{T-1}(\mathbf{x})$ *and* $\mu^*_{T-1}(\mathbf{x})$ *for a given* $\mathbf{x} \in \mathcal{X}$ *as*

$$\mathbb{P}\Big(|\mu^*_{T-1}(\mathbf{x}) - Q^*_{T-1}(\mathbf{x})| \leq \sqrt{\beta(\tau)}\sigma_{T-1}(\mathbf{x}, \mathbf{u}^*_{\mu_{T-1}}) + \gamma(\tau) + L_{\mu_{T-1}}|\mathbf{u}^*_{Q_{T-1}} - \mathbf{u}^*_{\mu_{T-1}}|, \quad \forall \mathbf{x} \in \mathcal{X}\Big) \geq 1 - \delta \tag{21}$$

In principle the analysis provided so far can apply to any system with a Lipschitz continuous cost function. However, when we propagate backwards, our analysis becomes limited to convex MDPs as in this case we preserve the Lipschitz continuity of the ground truth state-action value functions. We now proceed to introduce a probabilistic error bound and asymptotic convergence rates for the remaining time steps in the horizon.

Assumption 10 *For any time step* $k \in [t, T-1]$*, let* $\delta_k \in (0, 1)$ *be chosen such that* $\sum_{k=T-(t+1)}^{T-1} \delta_k \leq 1$.

Theorem 8 (Finite Horizon Optimal Value Function Error Bound) *Under the Assumptions above and by applying the union bound together with the error bound from Proposition 5 at each time step, we obtain the probabilistic error bound for discrete time index,* t*:*

$$\mathbb{P}\Big(\mid Q^*_t(\mathbf{x}_t) - \mu^*_t(\mathbf{x}_t) \mid \leq \sum_{k=t}^{T-1} \Big(\sqrt{\beta_k(\tau)}\sigma_k(\mathbf{x}_k, \mathbf{u}^*_k) + \gamma_k(\tau_k) + L_{\mu_k}|u^*_{Q_k} - u^*_{\mu_k}|\Big) \quad \forall \mathbf{x}_t \in \mathcal{X}\Big) \geq 1 - \sum_{k=t}^{T-1} \delta_k \tag{22}$$

where the state term, $\mathbf{x}_k, \forall k > t$ *within the standard deviation of the posterior process comprising the first term on the right hand side of the inequality within the probability operator, is yielded through forward evaluation of the decision process under the optimal policy generated from the posterior mean,* $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}^*_{\mu_k})$.

Theorem 8 provides a probabilistic uniform error bound between the optimal state-action value function and the Gaussian posterior process mean approximation for an available dataset.

Theorem 9 (Asymptotic Error Convergence) *Following from Theorem 8, as the number of data points* $N \to \infty$*, the posterior variance term* $\sigma_t(\cdot)$ *decreases as* $\mathcal{O}(1/\alpha(N))$ *uniformly over all timesteps* $k \in [t, T]$*, leading to an overall convergence rate of* $\mathcal{O}(\sqrt{\log(N)}/\alpha(N))$ *for the GP approximation error to the optimal Q-functions* $Q^*_k(\cdot)$ *across the entire time horizon.*

References

Jean-Pierre Aubin and Helene Frankowska. *Set-valued analysis*. Springer, 1999.

Kamyar Azizzadenesheli and Animashree Anandkumar. Efficient Exploration through Bayesian Deep Q-Networks, September 2019. URL http://arxiv.org/abs/1802.04412. arXiv:1802.04412 [cs].

Shane Barratt. On the differentiability of the solution to convex optimization problems. *arXiv preprint arXiv:1804.05098*, 2018.

Maximilian Bloor, José Torraca, Ilya Orson Sandoval, Akhil Ahmed, Martha White, Mehmet Mercangöz, Calvin Tsay, Ehecatl Antonio Del Rio Chanona, and Max Mowbray. Pc-gym: Benchmark environments for process control problems, 2024. URL https://arxiv.org/abs/2410.22093.

Stephen Boyd. Convex optimization. *Cambridge UP*, 2004.

Eric Bradford and Lars Imsland. Economic stochastic model predictive control using the unscented kalman filter. *IFAC-PapersOnLine*, 51(18):417–422, 2018. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

Eric Bradford, Lars Imsland, Dongda Zhang, and Ehecatl Antonio Del Rio Chanona. Stochastic data-driven model predictive control using gaussian processes. *Computers & Chemical Engineering*, 139:106844, August 2020. ISSN 0098-1354. DOI: 10.1016/j.compchemeng.2020.106844. URL https://linkinghub.elsevier.com/retrieve/pii/S0098135419313080. Publisher: Elsevier BV.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31, 2018.

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.

Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian Processes for Data-Efficient Learning in Robotics and Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, February 2015. ISSN 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2013.218. URL http://ieeexplore.ieee.org/document/6654139/.

Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pp. 201–208, Bonn, Germany, 2005. ACM Press. DOI: 10.1145/1102351.1102377. URL http://portal.acm.org/citation.cfm?doid=1102351.1102377.

Anthony V Fiacco and Yo Ishizuka. Sensitivity and stability analysis for nonlinear programming. *Annals of Operations Research*, 27(1):215–235, 1990.

Robert Grande, Thomas Walsh, and Jonathan How. Sample efficient reinforcement learning with gaussian processes. In *International Conference on Machine Learning*, pp. 1332–1340. PMLR, 2014.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012. ISSN 1935-8237, 1935-8245. DOI: 10.1561/2200000044. URL http://arxiv.org/abs/1207.6083. arXiv:1207.6083 [stat].

Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Malte Kuss and Carl Rasmussen. Gaussian processes in reinforcement learning. *Advances in neural information processing systems*, 16, 2003.

Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform error and posterior variance bounds for gaussian process regression with application to safe control. *CoRR*, abs/2101.05328, 2021a. URL https://arxiv.org/abs/2101.05328.

Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform error and posterior variance bounds for gaussian process regression with application to safe control. *arXiv preprint arXiv:2101.05328*, 2021b.

Ali Mesbah. Stochastic Model Predictive Control: An Overview and Perspectives for Future Research. *IEEE Control Systems*, 36(6):30–44, December 2016. ISSN 1066-033X, 1941-000X. DOI: 10.1109/mcs.2016.2602087. URL https://ieeexplore.ieee.org/document/7740982/. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12), 2006.

Henry B Moss, Sebastian W Ober, and Victor Picheny. Inducing point allocation for sparse gaussian processes in high-throughput bayesian optimisation. In *International Conference on Artificial Intelligence and Statistics*, pp. 5213–5230. PMLR, 2023.

M. Mowbray, P. Petsagkourakis, E.A. Del Rio-Chanona, and D. Zhang. Safe chance constrained reinforcement learning for batch process control. *Computers & Chemical Engineering*, 157:107630, January 2022. ISSN 0098-1354. DOI: 10.1016/j.compchemeng.2021.107630. URL https://linkinghub.elsevier.com/retrieve/pii/S0098135421004087. Publisher: Elsevier BV.

Thomas Savage, Dongda Zhang, Max Mowbray, and Ehecatl Antonio Del Río Chanona. Model-free safe reinforcement learning for chemical processes using Gaussian processes. *IFAC-PapersOnLine*, 54(3):504–509, 2021. ISSN 24058963. DOI: 10.1016/j.ifacol.2021.08.292. URL https://linkinghub.elsevier.com/retrieve/pii/S240589632101065X.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Rangarajan K Sundaram. *A first course in optimization theory*. Cambridge university press, 1996.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from gaussian process posteriors. In *International Conference on Machine Learning*, pp. 10292–10302. PMLR, 2020.

Supplementary Materials

The following content was not necessarily subject to peer review.

B Experimental Setup

B.1 Linear Quadratic Regulator

$$\min_{\mathbf{u}_0,\ldots,\mathbf{u}_{T-1}} \quad \mathbf{x}_T^{\mathsf{T}} Q_T \mathbf{x}_T + \sum_{k=0}^{T-1} (\mathbf{x}_k^T Q \mathbf{x}_k + \mathbf{u}_k^T R \mathbf{u}_k) \tag{23}$$

$$\text{s.t} \quad \mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{24}$$

$$\mathbf{x}_0 = \mathbf{x}(0) \tag{25}$$

$$\mathbf{u}_L \leq \mathbf{u}_k \leq \mathbf{u}_U \tag{26}$$

$$\forall k \in \mathcal{T} \setminus \{T\} \tag{27}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ represents the state vector, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the control input vector, $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ define the discrete-time system dynamics, $Q \in \mathbb{R}^{n_x \times n_x}$ and $R \in \mathbb{R}^{n_u \times n_u}$ are positive semi-definite and positive definite cost matrices for state and control respectively, $Q_N \in \mathbb{R}^{n_x \times n_x}$ is the terminal state cost matrix, $\mathbf{u}_L$ and $\mathbf{u}_U$ are the lower and upper control bounds respectively, and N denotes the finite time horizon. In the LQR experiment, the following dynamics matrices are used $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Cost matrices are set as $Q = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 0.1 \end{bmatrix}$ and $R = 0.1$.

B.2 Semi-Batch Reactor

The dynamics of the semi-batch reactor are shown below:

$$\frac{d\mathsf{C_A}}{dt} = \frac{\mathsf{F}}{\mathsf{V}}(\mathsf{C_{A,in}} - \mathsf{C_A}) - \mathsf{k_1 C_A} \tag{28}$$

$$\frac{d\mathsf{C_B}}{dt} = -\frac{\mathsf{F}}{\mathsf{V}}\mathsf{C_B} + \mathsf{k_1 C_A} - \mathsf{k_2 C_B} \tag{29}$$

$$\frac{d\mathsf{C_C}}{dt} = -\frac{\mathsf{F}}{\mathsf{V}}\mathsf{C_C} + \mathsf{k_2 C_B} \tag{30}$$

$$\frac{d\mathsf{T}}{dt} = \frac{\mathsf{UA(T_a - T) - F_{A0} C_{P_A} (T - T_b)}}{\mathsf{[C_A C_{P_A} + C_B C_{P_B} + C_C C_{P_C}]V + N_{H_2SO_4} C_{P_{H_2SO_4}}}} \tag{31}$$

$$+ \frac{\mathsf{[(\Delta H_{R1})(-k_1 C_A) + (\Delta H_{R2B})(-k_2 C_B)]V}}{\mathsf{[C_A C_{P_A} + C_B C_{P_B} + C_C C_{P_C}]V + N_{H_2SO_4} C_{P_{H_2SO_4}}}} \tag{32}$$

$$\frac{d\mathsf{V}}{dt} = \mathsf{F} \tag{33}$$

The variables in the system are denoted as follows: $\mathsf{C_A}$, $\mathsf{C_B}$, and $\mathsf{C_C}$ represent the concentrations (mol/dm^3) of species A, B, and C; T is the reactor temperature (K); V is the liquid volume (m^3); F is the flow rate of pure A entering the reactor (m^3/h); $\mathsf{T_a}$ is the temperature of the heat exchanger (K); $\mathsf{C_{P_A}}$, $\mathsf{C_{P_B}}$, $\mathsf{C_{P_C}}$ are the specific heat capacities of components A, B, and C; $\mathsf{\Delta H_{R1A}}$ and $\mathsf{\Delta H_{R2B}}$ are the reaction enthalpies for the first and second reactions; $\mathsf{k_1}$ and $\mathsf{k_2}$ are temperature-dependent rate constants.

The cost function is defined,

$$\Phi(\mathbf{x}, \mathbf{u}) = \begin{cases} -\mathsf{C_C}(t+1)\mathsf{V}(t+1) & \text{if } t = T-1 \\ 0 & \text{else} \end{cases}. \tag{34}$$