# Differentiating Policies for Non-Myopic Bayesian Optimization

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Bayesian optimization (BO) methods choose sample points by optimizing an acquisition function derived from a statistical model of the objective. These acquisition functions are chosen to balance sampling regions with predicted good objective values against exploring regions where the objective is uncertain. Standard acquisition functions are myopic, considering only the impact of the next sample, but non-myopic acquisition functions may be more effective. In principle, one could model the sampling by a Markov decision process, and optimally choose the next sample by maximizing an expected reward computed by dynamic programming; however, this is infeasibly expensive. More practical approaches, such as rollout, consider a parametric family of sampling policies. In this paper, we show how to efficiently estimate rollout acquisition functions and their gradients, enabling stochastic gradient-based optimization of sampling policies.

## 1 Introduction

Bayesian optimization (BO) algorithms are sample-efficient methods for global optimization of expensive continuous "black-box" functions for which we do not have derivative information. BO builds a statistical model of the objective based on samples, and chooses new sample locations by optimizing an acquisition function that reflects the value of candidate locations. Most BO methods use *myopic* acquisition functions that only consider the impact of the next sample. *Non-myopic* strategies [33] may produce high-quality solutions with fewer evaluations than myopic strategies. One can define an "optimal" non-myopic acquisition function in terms of the expected reward for a Markov decision process. But because of the curse of dimensionality, an exact dynamic programming approach to compute such a function is computationally intractable in almost any problem of interest. Approximate methods such as *rollout* are more practical, but remain expensive.

This paper aims to make non-myopic BO more practical. In particular, our main contributions are:

- We compute rollout acquisition functions via quasi-Monte Carlo integration and use variance reduction techniques to decrease the estimation error.
- We introduce a trajectory-based formulation for deriving non-myopic acquisition functions using myopic functions—or any heuristic—as base heuristics.
- We show how to differentiate rollout acquisition functions given a differentiable base policy.

We show on a suite of test problems that we tend to do better than myopic policies when considering the best decrease in objective function from the first iteration to the last.

The subsequent sections of this paper discuss background and related work, including Gaussian process regression, myopic Bayesian optimization, non-myopic Bayesian optimization, and rollout

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

acquisition functions (Section 2). Section 3 covers our models and methods, detailing our model and variance reduction techniques. We then explore our experimental results and their implications in Section 4. Limitations are discussed in Section 5. The paper concludes with our final thoughts in Section 6.

## 2 Background and related work

Non-myopic Bayesian optimization has received a lot of attention over the past few years [5; 25; 19; 17; 16; 6; 2; 3; 33]. Much of this work involves *rollout* methods, in which one considers the expected best value seen over $h$ steps of a standard (myopic) BO iteration starting from a sample point under consideration. Rollout acquisition functions represent state-of-the-art in BO and are integrals over $h$ dimensions, where the integrand itself is evaluated through inner optimizations, resulting in an expensive integral that is typically evaluated by Monte Carlo methods. The rollout acquisition function is then maximized to determine the next BO evaluation, further increasing the cost. This large computational overhead has been observed by Osborn et al. [25], who are only able to compute rollout acquisition for horizon 2, dimension 1. Lam et al. [17], who use Gauss-Hermite quadrature in horizons up to five, saw runtimes on the order of hours for small, synthetic functions [3].

Recent work focuses on making rollout more practical. Wu and Frazier [3] consider horizon 2, using a combination of Gauss-Hermite quadrature [20] and Monte Carlo (MC) integration to quickly calculate the acquisition function and its gradient. Non-myopic active learning also uses rollout [4; 9; 10; 15] and recent work develops a fast implementation by truncating the horizon and selecting a batch of points to collect future rewards [9; 10].

### 2.1 Gaussian process regression

A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution. Gaussian processes can be used to describe a distribution over functions. We place a GP prior on $f(\mathbf{x})$, denoted by $f \sim \mathcal{GP}(\mu, k)$, where $\mu : \Omega \to \mathbb{R}$ and $k : \Omega \times \Omega \to \mathbb{R}$ are the mean and covariance function, respectively. Here, $k$ is a kernel that correlates points in our sample space and it typically contains hyperparameters—like a lengthscale factor—that are learned to improve the quality of the approximation [28].

After observing $f$ at data points $\mathcal{D}_n = \{(\mathbf{x}^i, y_i) : 0 \leq i \leq n, i \in \mathbb{N}\}$, the posterior distribution on $f$ is a GP. We assume the observations are $y_i = f(\mathbf{x}^i) + \epsilon_i$ where the $\epsilon_i$ are independent variables distributed as $\mathcal{N}(0, \sigma^2)$. Given $\mathcal{D}_n$, we define the following for convenience of representation:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \mu_X = \begin{bmatrix} \mu(\mathbf{x}^1) \\ \vdots \\ \mu(\mathbf{x}^n) \end{bmatrix}, \quad \mathbf{k}(\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}, \mathbf{x}^1) \\ \vdots \\ k(\mathbf{x}, \mathbf{x}^n) \end{bmatrix}, \quad K = \begin{bmatrix} \mathbf{k}(\mathbf{x}^1)^T \\ \vdots \\ \mathbf{k}(\mathbf{x}^n)^T \end{bmatrix}.$$

The resulting posterior distribution for function values at a location $\mathbf{x}$ is the normal distribution $\mathcal{N}(\mu^{(n)}(\mathbf{x} \mid \mathcal{D}_n),\ K^{(n)}(\mathbf{x}, \mathbf{x} \mid \mathcal{D}_n))$:

$$\mu^{(n)}(\mathbf{x} \mid \mathcal{D}_n) = \mu(\mathbf{x}) + \mathbf{k}(\mathbf{x})^T (K + \sigma^2 I_n)^{-1} (\mathbf{y} - \mu_X) \tag{1}$$

$$K^{(n)}(\mathbf{x}, \mathbf{x} \mid \mathcal{D}_n) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T (K + \sigma^2 I_n)^{-1} \mathbf{k}(\mathbf{x}) \tag{2}$$

where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. GPs allow us to perform inference while quantifying our uncertainty. They also provide us with a mechanism for sampling possible belief states.

### 2.2 Myopic Bayesian optimization

Consider the problem of seeking a global minimum of a continuous objective $f(\mathbf{x})$ over a compact set $\Omega \subseteq \mathbb{R}^d$. If $f(\mathbf{x})$ is expensive to evaluate, then finding a minimum should be sample-efficient. Myopic Bayesian optimization (MBO) typically uses a GP to model $f(\mathbf{x})$ from the data $\mathcal{D}_n$. The next evaluation location $\mathbf{x}^{n+1}$ is determined by maximizing an acquisition function $\alpha(\mathbf{x} \mid \mathcal{D}_n)$ where the next sample is made based on a short-term view, looking only one step ahead:

$$\mathbf{x}^{n+1} = \arg\max_{\Omega} \alpha(\mathbf{x} \mid \mathcal{D}_n).$$

In the myopic setting, our sequential decision-making problem makes choices that depend on all past observations, optimizing for some immediate reward. However, we usually have more than one step in our budget, so myopic acquisition functions are leaving something behind.

### 2.3 Non-myopic Bayesian optimization

Non-myopic Bayesian optimization (NMBO) frames the exploration-exploitation problem as a balance of immediate and future rewards. Lam et al. [17] formulate NMBO as a finite horizon dynamic program; the equivalent Markov decision process follows.

The notation used is standard (see Puterman [26]): an MDP is a collection $(T, \mathbb{S}, \mathbb{A}, P, R)$, where $T = \{1, 2, \ldots, h\}$, and $h < \infty$ is the set of decision epochs, finite for our problem. The state space, $\mathbb{S}$, encapsulates the information needed to model the system from time $t \in T$, and $\mathbb{A}$ is the action space. Given a state $s \in \mathbb{S}$ and an action $a \in \mathbb{A}$, $P(s'|s, a)$ is the probability the next state will be $s'$. $R(s, a, s')$ is the reward received for choosing action $a$ in state $s$ and transitioning to $s'$.

A decision rule $\pi_t : \mathbb{S} \to \mathbb{A}$ maps states to actions at time $t$. A policy $\pi$ is a series of decision rules $\pi = (\pi_1, \pi_2, \ldots, \pi_h)$, one at each decision epoch. Given a policy $\pi$, a starting state $s_0$, and horizon $h$, we can define the expected total reward $V_h^\pi(s_0)$ as:

$$V_h^\pi(s_0) = \mathbb{E}\left[\sum_{t=0}^{h-1} R(s_t, \pi_t(s_t), s_{t+1})\right].$$

Our objective is to find the optimal policy $\pi^*$ that maximizes the expected total reward, i.e., $\sup_{\pi\in\Pi} V_h^\pi(s_0)$, where $\Pi$ is the space of all admissible policies.

If we can sample from the transition probability $P$, we can estimate the expected total reward of any base policy — the decisions made using the base acquisition function — with MC integration [32]:

$$V_h^{\hat{\pi}}(s_0) \approx \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{t=0}^{h-1} R(s_t^i, \hat{\pi}_t(s_t^i), s_{t+1}^i)\right].$$

Given a GP prior over data $\mathcal{D}_t$ with mean $\mu^{(t)}$ and covariance matrix $K^{(t)}$, we model $h$ steps of BO as an MDP. This MDP's state space is all possible data sets reachable from starting-state $\mathcal{D}_t$ with $h$ steps of BO. Its action space is $\Omega$; actions correspond to sampling a point in $\Omega$. Its transition probability and reward function are defined as follows. Given an action $x^{t+1}$, the transition probability from $\mathcal{D}_t$ to $\mathcal{D}_{t+1}$, where $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\mathbf{x}^{t+1}, y_{t+1})\}$ is:

$$P(\mathcal{D}_t, \mathbf{x}^{t+1}, \mathcal{D}_{t+1}) \sim \mathcal{N}(\mu^{(t)}(\mathbf{x}^{t+1}; \mathcal{D}_t), K^{(t)}(\mathbf{x}^{t+1}, \mathbf{x}^{t+1}; \mathcal{D}_t)).$$

Thus, the transition probability from $\mathcal{D}_t$ to $\mathcal{D}_{t+1}$ is the probability of sampling $y_{t+1}$ from the posterior $\mathcal{GP}(\mu^{(t)}, K^{(t)})$ at $\mathbf{x}^{t+1}$. We define a reward according to expected improvement (EI) [12]. Let $f_t^*$ be the minimum observed value in the observed set $\mathcal{D}_t$, i.e. $f_t^* = \min\{y_0, \ldots, y_t\}$. Then our reward is expressed as follows:

$$R(\mathcal{D}_t, \mathbf{x}^{t+1}, \mathcal{D}_{t+1}) = (f_t^* - f_{t+1})^+ \equiv \max(f_t^* - f_{t+1}, 0).$$

EI can be defined as the optimal policy for horizon one, obtained by maximizing the immediate reward:

$$\pi_{EI} = \arg\max_\pi V_1^\pi(\mathcal{D}_t) = \arg\max_{\mathbf{x}^{t+1}\in\Omega} \mathbb{E}\left[(f_t^* - f_{t+1})^+\right] \equiv \arg\max_{\mathbf{x}^{t+1}\in\Omega} EI(\mathbf{x}^{t+1}|\mathcal{D}_t),$$

where the starting state is $\mathcal{D}_t$—our initial samples. We define the non-myopic policy as the optimal solution to an $h$-horizon MDP. The expected total reward of this MDP is:

$$V_h^\pi(\mathcal{D}_n) = \mathbb{E}\left[\sum_{t=n}^{n+h-1} R(\mathcal{D}_t, \pi_t(\mathcal{D}_t), \quad \mathcal{D}_{t+1})\right] = \mathbb{E}\left[\sum_{t=n}^{n+h-1} (f_t^* - f_{t+1})^+\right] = \mathbb{E}\left[\left(f_n^* - \min_{t\le n+h} f_t\right)^+\right].$$

The integrand is a telescoping sum, resulting in the equivalent expression on the rightmost side of the equation. When $h > 1$, the optimal policy is difficult to compute.

### 2.4 Rollout acquisition functions

Finding an optimal policy for the $h$-horizon MDP is hard, partly because of the expense of representing and optimizing over policy functions. One approach to faster approximate solutions to the MDP is to replace the infinite-dimensional space of policy functions with a family of trial policies described by a modest number of parameters. An example of this approach is *rollout* policies [1], which yield promising results for the $h$-step horizon MDP [3]. For a given state $\mathcal{D}_t$, we say that a base policy $\tilde{\pi}$ is a sequence of rules that determines the actions to be taken in various states of the system.

We denote our base policy for the $h$-horizon MDP as $\tilde{\pi} = (\tilde{\pi}_0, \tilde{\pi}_1, \tilde{\pi}_2, \ldots, \tilde{\pi}_h)$. We let $\mathcal{D}_t$ denote the initial state of our MDP and $\mathcal{D}_{t,k}$ for $0 \leq k \leq h$ to denote the random variable that is the state at each decision epoch. Each individual decision rule $\tilde{\pi}_k$ for $k \geq 1$ consists of maximizing the base acquisition function $\bar{\alpha}$ given the current state $s_k = \mathcal{D}_{t,k}$, i.e.

$$\tilde{\pi}_k = \underset{\mathbf{x} \in \Omega}{\arg\max}\, \bar{\alpha}\left(\mathbf{x} | \mathcal{D}_{t,k}\right).$$

Using this policy, we define the non-myopic acquisition function $\alpha_h(\mathbf{x})$ as the rollout of $\tilde{\pi}$ to horizon $h$, i.e. the expected reward of $\tilde{\pi}$ starting with action $\tilde{\pi}_0 = \mathbf{x}^{\text{start}}$:

$$\alpha_h\left(\mathbf{x}^{\text{start}}\right) \coloneqq \mathbb{E}\left[V_h^{\tilde{\pi}}\left(\mathcal{D}_t \cup \left\{\left(\mathbf{x}^{\text{start}}, y_{\text{start}}\right)\right\}\right)\right],$$

where $y_{\text{start}}$ is the noisy observed value of $f$ at $\mathbf{x}^{\text{start}}$. Thus, as is the case with any acquisition function, the next BO evaluation is:

$$\mathbf{x}^{\text{next}} = \underset{\mathbf{x}^{\text{start}} \in\, \Omega}{\arg\max}\, \alpha_h\left(\mathbf{x}^{\text{start}}\right).$$

Rollout is tractable and conceptually straightforward, however, it is still computationally demanding. To rollout $\tilde{\pi}$ once, we must do $h$ steps of BO with $\bar{\alpha}$. Many of the aforementioned rollouts must then be averaged to reasonably estimate $\alpha_h$, which is an $h$-dimensional integral. Estimation can be done either through explicit quadrature or MC integration, and is the primary computational bottleneck of rollout.

## 3 Models and methods

### 3.1 Model

We build our intuition behind our approach from a top-down perspective. We have seen that non-myopic Bayesian optimization is promising, though tends to be computationally intractable. To circumvent this problem, we formulate a sub-optimal approximation to solve the intractable dynamic program; namely, a rollout acquisition function. Though relatively more tractable, rollout acquisition functions can be computationally burdensome. We are interested in solving $x^* = \arg\max_{x \in \mathcal{X}} \alpha_h(x)$ where

$$\alpha_h(\mathbf{x}) = \mathbb{E}_{\hat{f}^i \sim \mathcal{G}}\left[\alpha\big(\mathbf{x} | \tau(h, \mathbf{x}, \bar{\alpha}, \hat{f}^i)\big)\right] \approx \frac{1}{N}\sum_{i=1}^{N}(f^* - \big(t^i(\mathbf{x})\big)^-)^+ \tag{3}$$

and $t^i(x) \sim \tau(h, x, \bar{\alpha}, \hat{f}^i)$ are sample trajectories. Sample trajectories consist of picking a starting point and asking "what would happen if I used my base policy from this point forward?" Since these trajectories don't represent what did happen, rather what might, we call each realization a fantasized sample. Unfortunately, derivative-free optimization in high-dimensional spaces is expensive, so we would like estimates of $\nabla_x \alpha_h(x)$. In particular, differentiating with respect to $x$ yields the following:

$$\nabla_x\left[\alpha_h(\mathbf{x})\right] \approx \nabla_x\left[\frac{1}{N}\sum_{i=1}^{N}(f^* - \big(t^i(\mathbf{x})\big)^-)^+\right] = \frac{1}{N}\sum_{i=1}^{N}\left(-\nabla_x\left[\big(t^i(\mathbf{x})\big)^-\right]\right)^+ \tag{4}$$

which requires some notion of differentiating sample trajectories. Thus, our problem can be expressed as two interrelated optimizations:

1. An inner optimization for determining sample trajectories $t^i(\mathbf{x}) \sim \tau(h, \mathbf{x}, \bar{\alpha}, \hat{f}^i)$
2. An outer optimization for determining our next query location $\mathbf{x}^*$

In what follows, we define how to compute and differentiate sample trajectories and how to differentiate the rollout acquisition function. We introduce the following notation to distinguish between two GP sequences that must be maintained. Suppose we have function values and gradients denoted as follows: $(\hat{f}_0, \nabla\hat{f}_0), \ldots (\hat{f}_h, \nabla\hat{f}_h)$. We define two GP sequences as:

$$\begin{aligned}\mathcal{F}_h \;&\sim\; \mathcal{F}_0 \mid \mathcal{F}_h(\mathbf{x}^0) = \hat{f}_0, \ldots, \mathcal{F}_h(\mathbf{x}^h) = \hat{f}_h \\ \mathcal{G}_h \;&\sim\; \mathcal{F}_0 \mid \mathcal{F}_h(\mathbf{x}^0) = \hat{f}_0, \ldots, \mathcal{F}_h(\mathbf{x}^h) = \hat{f}_h, \\ &\qquad \nabla\mathcal{F}_h(\mathbf{x}^0) = \nabla\hat{f}_0, \ldots, \nabla\mathcal{F}_h(\mathbf{x}^h) = \nabla\hat{f}_h.\end{aligned}$$

The model, fundamentally, relies on the distinction amongst known sample locations $\{\mathbf{x}^{-j} \in \mathbb{R}^d \mid 1 \leq j \leq m\}$, deterministic start location $\{\mathbf{x}^0 \in \mathbb{R}^d\}$, and the stochastic fantasized sample locations $\{\mathbf{x}^j \in \mathbb{R}^d \mid 1 \leq j \leq h\}$. We denote a full trajectory including $h$ fantasized steps by:

$$X^{m+h+1} := \left[\mathbf{x}^{-m} \;\ldots\; \mathbf{x}^{-1} \middle|\, \mathbf{x}^0 \middle|\, \mathbf{x}^1 \;\ldots\; \mathbf{x}^h\right] \in \mathbb{R}^{d\times(m+h+1)}.$$

Moreover, we collect the $m$ known samples into $y = [y_{-m} \;\ldots\; y_{-1}]^T$. The distinction between negative, null, and positive superscripts serves a useful purpose. Negative superscripts can be thought of as the past; potentially noisy observations that are fixed and immutable. The null superscripts denotes our freedom of choice; we are not bound to start at a specific location. Positive superscripts can be thought of as our ability to see into the future: things we anticipate on observing given our current beliefs.

Our problem is how to choose $\mathbf{x}^0$ to maximize our expected reward over some finite horizon $h$. We focus on developing an $h$-step expected improvement (EI) rollout policy that involves choosing $\mathbf{x}^0$ by using Stochastic Gradient Ascent (SGA) to optimize our rollout acquisition function.

An $h$-step EI *rollout* policy involves choosing $\mathbf{x}^0$ based on the anticipated behavior of the EI ($\bar{\alpha}$) algorithm starting from $\mathbf{x}^0$ and proceeding for $h$ steps. That is, we consider the iteration

$$\mathbf{x}^r = \arg\max_x \bar{\alpha}(\mathbf{x} \mid \mathcal{F}_{r-1}),\; 1 \leq r \leq h \tag{5}$$

where the trajectory relations are

$$\nabla_x\bar{\alpha}(\mathbf{x}^r|\mathcal{F}_{r-1}) = \mathbf{0} \;\wedge\; \left(\hat{f}_r, \nabla\hat{f}_r\right) \sim \mathcal{G}_{r-1}(\mathbf{x}^r).$$

Trajectories are fundamentally random variables; their behavior is determined by the rollout horizon $h$, start location $\mathbf{x}$, base policy $\bar{\alpha}$, and initial observation $\hat{f}^i$—denoted as $\tau(h, \mathbf{x}, \bar{\alpha}, \hat{f}^i)$. We denote sample draws $t^i \sim \tau(h, \mathbf{x}, \bar{\alpha}, \hat{f}^i)$ as follows:

$$t^i = \left(\left(\mathbf{x}^j, \hat{f}^i_j(\mathbf{x}^j), \nabla\hat{f}^i_j(\mathbf{x}^j)\right)\right)_{j=0}^{h} \tag{6}$$

Computing this sample path, however, requires we solve the iteration defined above. We also use the notation $t^i_{j,k}$ to denote the $k$-th element of the $j$-th 3-tuple associated with the $i$-th sample. Now that we have some provisional notation, we define how to evaluate $t^i(x)$ as follows:

$$(t^i(\mathbf{x}))^- \equiv \min(t^i(\mathbf{x})) \equiv \min_{0\leq j\leq h} t^i_{j,2} \tag{7}$$

If we let $b = \arg\min_j t^i_{j,2}$, we can rewrite the minimum of the sample trajectory as follows:

$$(t^i(\mathbf{x}))^- \equiv \hat{f}^i_b(\mathbf{x}^b(\mathbf{x})). \tag{8}$$

Now, we are able to differentiate trajectories given by the following

$$\nabla_x\left[(t^i(\mathbf{x}))^-\right] = \left[\tfrac{\partial}{\partial x_1}\left[(t^i(\mathbf{x}))^-\right],\; \ldots,\; \tfrac{\partial}{\partial x_d}\left[(t^i(\mathbf{x}))^-\right]\right]^T \tag{9}$$

which requires us to compute $\frac{\partial}{\partial x_k}\left[(t^i(\mathbf{x}))^-\right]$ where $1 \leq k \leq d$. Hence, we have

$$\frac{\partial}{\partial x_k}\left[(t^i(\mathbf{x}))^-\right] = \frac{\partial}{\partial x_k}\left[\hat{f}_b(\mathbf{x}^b(\mathbf{x}))\right]. \tag{10}$$

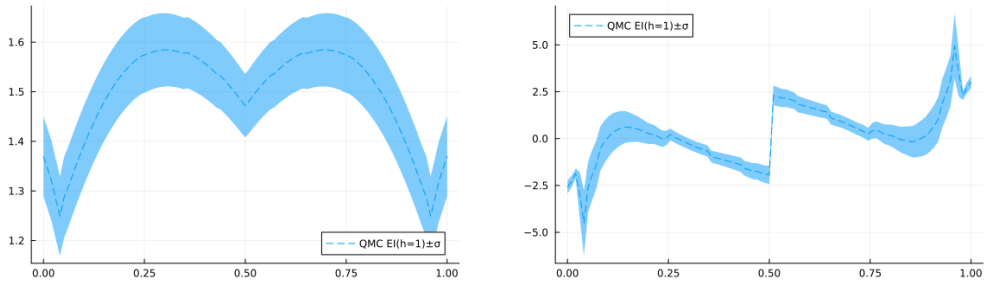


Figure 1: The first graph is the rollout acquisition function for a non-trivial horizon. Subsequently, we depict the estimated gradient and their respective standard errors.

Since the best function value $\hat{f}_b$ found is a function of the best location $x^b$, which is also a function of $x$, we have

$$\frac{\partial}{\partial x_k}\left[\hat{f}_b(\mathbf{x}^b(\mathbf{x}))\right] = \frac{\partial \hat{f}_b}{\partial \mathbf{x}^b}\frac{\partial \mathbf{x}^b}{\partial x_k}. \tag{11}$$

This captures how the best value found thus far changes as we vary the $k$-th dimension of $x$ which is equivalently expressed as

$$\nabla_x\left[\hat{f}_b(\mathbf{x}^b(\mathbf{x}))\right] = \left(\hat{f}_b'(\mathbf{x}^b(\mathbf{x})) \cdot \frac{\partial \mathbf{x}^b}{\partial x}\right)^T. \tag{12}$$

From the trajectory relations defined above, we are able to derive the following relationship via the Implicit Function Theorem about $x^r$:

$$\frac{\partial \bar{\alpha}}{\partial \mathbf{x}^r}(\mathbf{x}^r|\mathcal{F}_{r-1}) = \mathbf{0} \rightarrow \frac{\partial^2 \bar{\alpha}}{(\partial \mathbf{x}^r)^2}\frac{\partial \mathbf{x}^r}{\partial \mathbf{x}} + \frac{\partial^2 \bar{\alpha}}{\partial \mathbf{x}^r \partial \mathcal{F}_{r-1}}\frac{\partial \mathcal{F}_{r-1}}{\partial \mathbf{x}} = \mathbf{0}.$$

We are able to compute the Jacobian matrix $J(\mathbf{x}^b) = \frac{\partial \mathbf{x}^b}{\partial \mathbf{x}}$ as

$$\frac{\partial \mathbf{x}^b}{\partial \mathbf{x}} = -\left(\frac{\partial^2 \bar{\alpha}}{(\partial \mathbf{x}^b)^2}\right)^{-1}\left(\frac{\partial^2 \bar{\alpha}}{\partial \mathbf{x}^b \partial \mathcal{F}_{b-1}}\frac{\partial \mathcal{F}_{b-1}}{\partial \mathbf{x}}\right) \tag{13}$$

defining all of the computations necessary to solve the inner and outer optimization problem.

### 3.2 Methods

Significant computational challenges arise when computing the rollout acquisition function (RAF). Namely, rolling out the base acquisition function for $h$ steps consists of updating and maintaining a fantasized GP which is averaged across some $N$ monte-carlo simulations. Naive updates of the covariance and Cholesky matrix for each fantasized GP unnecessarily repeats computations. Another cost incurred when evaluating the rollout acquisition function (RAF) is that the $h$-step iteration is solving an optimization problem at each step. Finally, in order to differentiate the RAF, we need to differentiate trajectories, requiring us to estimate a sequence of Jacobian matrices. In what follows, we address the following techniques to enable efficient computation and optimization of the RAF: smart linear algebra, quasi-monte carlo, common random numbers and control variates.

**Differentiation of the base acquisition function** Evaluating the rollout acquisition function necessitates the optimization of the base acquisiton function iteratively, for a total of $h$ times. To expedite this step, our approach uses the gradient and Hessian of the base policy. This enables us to use second-order optimization algorithms that significantly speeds up the optimization process.

**Smart linear algebra.** Maintaining $N$ GPs with an $h$-step base policy for some evaluation of $\alpha_h$ at location $x^0$ is expensive. The value of $h$ determines how many inner optimization problems we solve and we call $\alpha_h(x^0)$ the $(h+1)$-step lookahead acquisition function. Computing a covariance

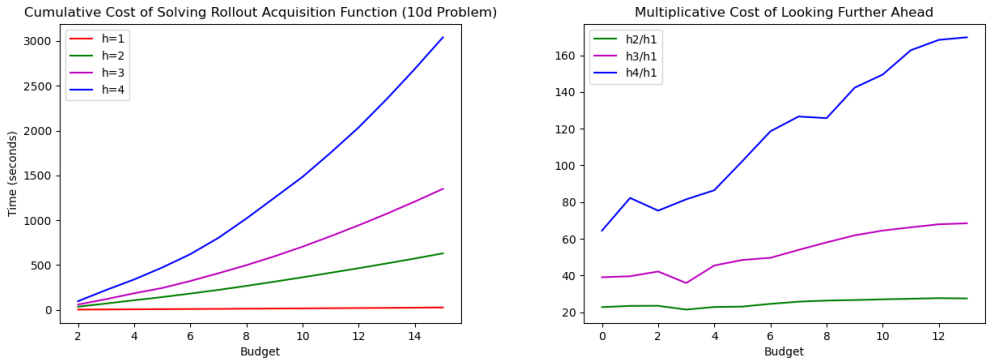


Figure 2: The first graph demonstrates the cost of increasing your horizon as your iterations progress. Subsequently, we depict the multiplicative cost of looking further ahead relative to the rollout acquisition function with $h = 1$. Note, $h = 1$ corresponds to a two-step lookahead strategy.

matrix and it's inverse is $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$, respectively. Doing this for our $h$-step policy produces something on the order of $\sum_{i=0}^{h} \mathcal{O}((n+i)^2)$ for the covariance updates and $\sum_{i=0}^{h} \mathcal{O}((n+i)^3)$ for the matrix inversion updates. To circumvent this inefficiency, we preallocate matrices for both the covariance and Cholesky factorizations. This preallocation is dynamically updated using Schur's complement. This approach yields a substantial reduction in computational load by making updates something on the order of a polynomial of a lower integer degree, thereby streamlining the inner optimization process.

### 3.2.1 Variance reduction techniques

Variance reduction is a set of methods that improve convergence by decreasing the variance of some estimator. Useful variance reduction methods can reduce the sample variance by several orders of magnitude [24]. We use a combination of quasi-Monte Carlo, common random numbers, and control variates, which significantly reduces the number of MC samples needed, while smoothing out estimates of the underlying stochastic objective.

**Quasi-monte carlo. (QMC)** Monte Carlo methods use independent, uniformly distributed random numbers on the $p$-dimensional unit cube as the source of points to integrate at. The distribution we integrate over is Normal, for which a low-discrepancy sequence is known to exist. We generate low-discrepancy Sobolev sequences in the $p$-dimensional uniform distribution $\mathcal{U}[0,1]^p$ and map them to the standard multivariate Gaussian via the Box-Muller transform [31]. This produces a low-discrepancy sequence for $\mathcal{N}(0, I_p)$, allowing us to apply QMC by replacing the standard multivariate Gaussian samples with our low-discrepancy sequence.

**Common random numbers (CRN).** In many stochastic processes, different iterations or simulations use separate sets of random numbers to generate outcomes. While this approach is robust in ensuring the independence of each trial, it can also introduce a high degree of variability or noise into the results. This is where Common Random Numbers come into play. The CRN techniques involves using the same set of random numbers across different iterations. CRN does not decrease the pointwise variance of an estimate, but rather decreases the covariance between two neighboring estimates, which smooths out the function.

**Control variates.** A control variate is a variable that is highly correlated with the function of interest and has a known expected value. The key is to find a variable whose behavior is similar to that of the function being estimated. Since $\alpha_h(x) = \mathbb{E}\left[\hat{\alpha}_h(x)\right]$ corresponds to an $h$-step EI acquisition, we use the 1-step EI acquisition as our control variate. That is, we can estimate $\alpha_h^{\text{cv}}(x) = \mathbb{E}\left[\hat{\alpha}_h(x) + \beta w(x)\right]$ where $w(x) = \max(f^+ - f(x), 0) - \text{EI}(x)$. Given this construction, the variance of our estimator is given as follows:

$$\begin{aligned}\text{Var}\left(\alpha_h^{\text{cv}}(x)\right) &= \text{Var}\left(\hat{\alpha}_h(x) + \beta w(x)\right) \\ &= \text{Var}\left(\hat{\alpha}_h(x)\right) + \beta^2 \text{Var}\left(w(x)\right) + 2\beta \text{Cov}\left(\hat{\alpha}_h(x), w(x)\right).\end{aligned}$$

The variance of our estimator is quadratic in $\beta$ and is minimized for $\beta^* = \frac{-\text{Cov}(\hat{\alpha},\text{w})}{\text{Var}(w)}$. However, since $\text{Cov}(\hat{\alpha},\text{w})$ is unknown, we estimate $\beta^*$ from the Monte Carlo simulations.

Overall, by combining smart linear algebra, quasi-Monte Carlo integration, common random numbers, and control variates, we show that computing and differentiating rollout acquisition functions is tractable.

# 4 Experiments and discussion

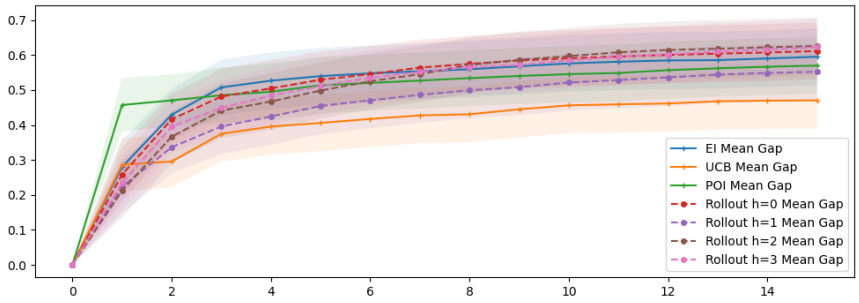


Figure 3: The average GAP and standard error across a suite of 15 synthetic test functions with 60 randomized trials and a single random observation as an initialization of the underlying GP model. All nonmyopic strategies outperform POI and most outperform EI.

Throughout our experiments we use a GP with the Matérn 5/2 ARD kernel [30] and learn its hyperparameters via maximum likelihood estimation [27]. When rolling out acquisition functions, we maximize them using the Adam-variant of Stochastic Gradient Ascent (SGA) with expected improvement used as the base policy. We use the proposed default values for Adam of 0.9 for $\beta_1$, 0.999 for $\beta_2$, and $10^{-8}$ for $\epsilon$ [14]. We use the evidence-based criterion developed in [22] as our convergence criteria. We use 8 random restarts for solving the inner optimization with a maximum of 50 iterations for SGA and select the best point found. All synthetic functions are found in the supplementary.

Given a fixed evaluation budget, we evaluate the performance of an algorithm in terms of its gap $G$. The gap measures the best decrease in the objective function from the first to the last iteration, normalized by the maximum reduction possible:

$$G = \frac{f_{\text{min}}^{\mathcal{D}_1} - f_{\text{min}}^{\mathcal{D}_{B+1}}}{f_{\text{min}}^{\mathcal{D}_1} - f(x^*)}$$

Across a suite of synthetic test functions, the non- myopic strategies were favored slightly more than the myopic strategies. The mean gap and standard error over 60 randomized trials with different initial observations are plotted in Fig. 4 for various myopic and non-myopic acquisition functions. Provided the base policy is a sequentially improving heuristic, [33] show that rollout is guaranteed to outperform its counterpart. In fact, Table 1 shows the median statistic for all non-myopic strategies outperforms the myopic counterpart of the $h$-step EI acquisition function, that is EI.

# 5 Limitations

The number of experiments conducted was limited due to hardware considerations. The experiments were run on a Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz Linux workstation with 8 CPU cores. We did not treat some important considerations that might yield better results. Namely, our MDP model does not consider a discount factor, which may yield different behaviors. When solving both the myopic and non-myopic acquisition functions, our tolerance in $\mathbf{x}^{\text{opt}}$ is $1\text{e}{-3}$. Using the implicit function theorem, we estimate the sequence of Jacobians using forward-mode differentiation, introducing a non-trivial expense. We also use the heuristic of $200h$ iterations when evaluating $\alpha_h(\mathbf{x})$, mimicking this choice from prior work done in [18].

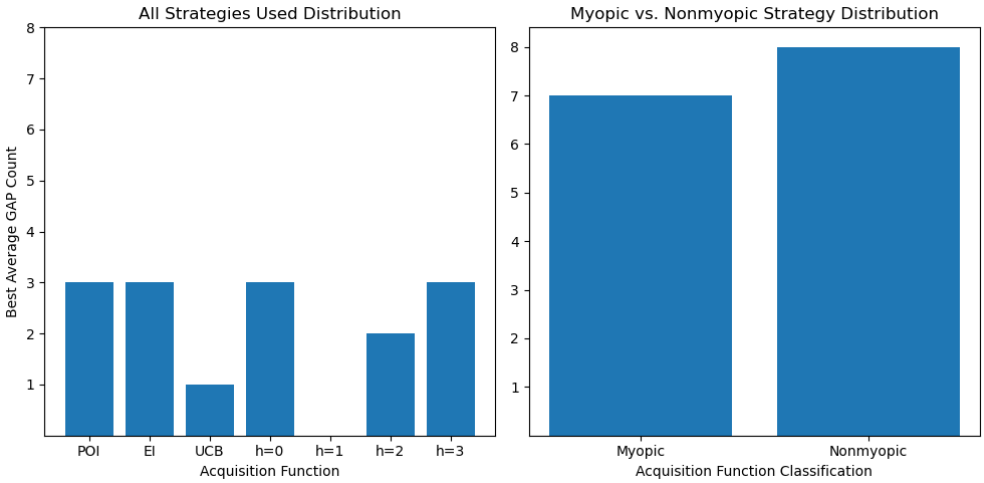


Figure 4: A histogram depicting the average winning strategies.

Table 1: Mean and median gaps for various functions and methods ran for 15 iterations of BO over 60 initial guesses.

| Function name | | PI | EI | UCB | KG | $\alpha_0$ | $\alpha_1$ | $\alpha_2$ | $\alpha_3$ |
|---|---|---|---|---|---|---|---|---|---|
| Gramacylee | Mean | .590 | .594 | .268 | .608 | .750 | .665 | .842 | **.875** |
| | Median | .585 | .631 | .181 | .700 | .796 | .707 | **.997** | .991 |
| Schwefel4d | Mean | .074 | .075 | .124 | **.257** | .073 | .140 | .137 | .163 |
| | Median | .015 | .017 | .017 | **.262** | 0.0 | .124 | .090 | .057 |
| Rosenbrock | Mean | .606 | .435 | .425 | **.874** | .813 | .683 | .594 | .597 |
| | Median | .806 | .431 | .403 | **.975** | .954 | .885 | .845 | .847 |
| Branin-Hoo | Mean | .440 | .533 | .318 | **.888** | .506 | .507 | .550 | .490 |
| | Median | .438 | .582 | 0.0 | **.947** | .576 | .623 | .695 | .598 |
| Goldstein-Price | Mean | .770 | .712 | .099 | **.928** | .818 | .784 | .770 | .742 |
| | Median | .932 | .877 | 0.0 | **.983** | .959 | .892 | .939 | .935 |
| Six-Hump Camel | Mean | .808 | .726 | .279 | .823 | .799 | .768 | .910 | **.934** |
| | Median | .892 | .861 | 0.0 | .934 | .917 | .874 | .961 | **.965** |

# 6 Conclusion

Our insights yield several interesting research directions. We have shown how to construct and optimize over a parametric family of viable policy functions, also known as a policy gradient method, using well-known myopic acquisition functions. One might be interested in treating the exploration-exploitation tradeoff as a parameter to be learned via gradient ascent as well, which our framework enables, suggesting another interesting research direction. This formulation also enables the treatment of the horizon as an adaptive hyperparameter learned as one iterates. Furthermore, when comparing one-step against multi-step lookahead strategies using the same base acquisition function, we observe that there is no definitive best strategy. This suggest that a finite horizon should not be fixed for the entire iteration.

Our contributions focused on using expected improvement as our base policy, but our analysis is base policy agnostic and only assumes the given base policy has two derivatives. If one has access to a fast automatic differentiation compute kernel for any base policy, this would enable the use of arbitrary base heuristics within the proposed framework. A natural next step would be to study similar non-myopic constructions using existing myopic acquisition functions while also considering the applying this methodology to real-world optimization problems where Bayesian optimization excels.

# References


[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, Belmont, MA, USA, 3rd edition, 2005.

[2] P. I. Frazier. A Tutorial on Bayesian Optimization. in(Section 5):1–22, 2018.

[3] P. I. Frazier. Practical Two-Step Look-Ahead Bayesian Optimization. (NeurIPS), 2019.

[4] R. Garnett, Y. Krishnamurthy, X. Xiong, J. Schneider, and R. Mann. Bayesian optimal active search and surveying. *arXiv preprint arXiv:1206.6406*, 2012.

[5] D. Ginsbourger and R. Le Riche. Towards Gaussian Process-based Optimization with Finite Time Horizon. (Umr 5146):89–96, 2010.

[6] J. González, M. Osborne, and N. D. Lawrence. GLASSES: Relieving the myopia of Bayesian optimisation. *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016*, 41:790–799, 2016.

[7] J. C. Goodson, B. W. Thomas, and J. W. Ohlmann. A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. *European Journal of Operational Research*, 258(1):216–229, 2017.

[8] R. R. Griffiths and J. M. Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2):577–586, 2020.

[9] S. Jiang, G. Malkomes, G. Converse, A. Shofner, B. Moseley, and R. Garnett. Efficient nonmyopic active search. In *International Conference on Machine Learning*, pages 1714–1723. PMLR, 2017.

[10] S. Jiang, G. Malkomes, M. Abbott, B. Moseley, and R. Garnett. Efficient nonmyopic batch active search. *Advances in Neural Information Processing Systems*, 31, 2018.

[11] S. Jiang, D. R. Jiang, M. Balandat, B. Karrer, J. R. Gardner, and R. Garnett. Efficient nonmyopic bayesian optimization via one-shot multi-step trees. *Advances in Neural Information Processing Systems*, 2020-Decem(NeurIPS), 2020.

[12] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient Global Optimization of Expensive Black-Box Functions," , vol. 13, no. 4, pp. 455-492, 1998. *Journal of Global Optimization*, 13:455–492, 1998.

[13] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 1996.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[15] A. Krause and C. Guestrin. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *Proceedings of the 24th international conference on Machine learning*, pages 449–456, 2007.

[16] R. R. Lam and K. E. Willcox. Lookahead Bayesian optimization with inequality constraints. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):1891–1901, 2017.

[17] R. R. Lam, K. E. Willcox, and D. H. Wolpert. Bayesian optimization with a finite budget: An approximate dynamic programming approach. *Advances in Neural Information Processing Systems*, (Nips):883–891, 2016.

[18] E. H. Lee, D. Eriksson, B. Cheng, M. McCourt, and D. Bindel. Efficient Rollout Strategies for Bayesian Optimization. 2020.

[19] C. K. Ling, K. H. Low, and P. Jaillet. Gaussian process planning with lipschitz continuous reward functions: Towards unifying Bayesian optimization, active learning, and beyond. *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pages 1860–1866, 2016.

[20] Q. Liu and D. A. Pierce. A note on gauss-hermite quadrature. *Biometrika*, 81(3):624–629, 1994. ISSN 00063444. URL `http://www.jstor.org/stable/2337136`.

[21] A. Mahajan and D. Teneketzis. *Multi-Armed Bandit Problems*. Number February 2014. 2008.

[22] M. Mahsereci, L. Balles, C. Lassner, and P. Hennig. Early Stopping without a Validation Set. pages 1–16, 2017. URL `http://arxiv.org/abs/1703.09580`.

[23] J. Mockus. *Bayesian Approach to Global Optimization*. Springer, 1989.

[24] O. Opc. Variance Reduction. 348(c):4–9, 1993.

[25] M. A. Osborne, R. Garnett, and S. J. Roberts. Gaussian Processes for Global Optimization. *3rd International Conference on Learning and Intelligent Optimization LION3*, (x):1–15, 2009.

[26] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[27] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[28] C. E. Rasmussen and W. C. K. I. *Gaussian processes for machine learning*. MIT Press, 2006.

[29] S. Sethi and G. Sorger. A theory of rolling horizon decision making. *Annals of Operations Research*, 29 (1):387–415, 1991.

[30] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, 4:2951–2959, 2012.

[31] R. S. Society. The Box-Cox Transformation Technique : A Review Author ( s ): R . M . Sakia Source : Journal of the Royal Statistical Society . Series D ( The Statistician ) , 1992 , Vol . Published by : Wiley for the Royal Statistical Society Stable URL : https://www.jstor.org/stable/2348250 REFERENCES Linked references are available on JSTOR for this article : reference # references _ tab _ contents You may need to log in to JSTOR to access the linked references . y ( A ) = Yt This content downloaded from. 41(2): 169–178, 1992.

[32] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL `http://incompleteideas.net/book/the-book-2nd.html`.

[33] X. Yue and R. A. Kontar. Why Non-myopic Bayesian Optimization is Promising and How Far Should We Look-Ahead? A Study via Rollout. 108, 2019.

# A Appendix / supplemental material

# B Kernels

The kernel functions used in this paper is the Matérn 5/2 kernel:

$$K_{5/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{5}}{\ell} + \frac{5}{3\ell^2}\right) \exp(-\frac{\sqrt{5}r}{\ell})$$

# C Expected Improvement

$$\begin{aligned}
\alpha(x) &= \sigma(x) g(z(x)) \\
g(z) &= z\Phi(z) + \phi(z) \\
z(x) &= \sigma(x)^{-1} \left[\mu(x) - f^+ - \xi\right]
\end{aligned}$$

# D Differentiating $\alpha$

Differentiating with respect to spatial coordinates:

$$\begin{aligned}
\alpha_{,i} &= \sigma_{,i} g(z) + \sigma g'(z) z_{,i} \\
\alpha_{,ij} &= \sigma_{,ij} g(z) + \sigma_{,i} g'(z) z_{,j} + \sigma_{,j} g'(z) z_{,i} + \sigma g'(z) z_{,ij} + \sigma g''(z) z_{,i} z_{,j} \\
&= \sigma_{,ij} g(z) + [\sigma_{,i} z_{,j} + \sigma_{,j} z_{,i} + \sigma z_{,ij}] g'(z) + \sigma g''(z) z_{,i} z_{,j}
\end{aligned}$$

Mixed derivative with respect to spatial coordinates and data and hypers:

$$\dot{\alpha}_{,i} = \dot{\sigma}_{,i} g(z) + \sigma_{,i} g'(z) \dot{z} + \dot{\sigma} g'(z) z_{,i} + \sigma g''(z) \dot{z} z_{,i} + \sigma g'(z) \dot{z}_{,i}$$

Finally, we differentiate $g(z) = z\Phi(z) + \phi(z)$, noting that $\phi'(z) = -z\phi(z)$ and $\Phi'(z) = \phi(z)$. This gives

$$\begin{aligned}
g(z) &= z\Phi(z) + \phi(z) \\
g'(z) &= \Phi(z) + z\phi(z) + \phi'(z) = \Phi(z) \\
g''(z) &= \phi(z).
\end{aligned}$$

# E Differentiating $z$

Now consider $z = \sigma^{-1}[\mu - f^+ - \xi]$. As before, we begin with spatial derivatives:

$$\begin{aligned}
z_{,i} &= -\sigma^{-2} \sigma_{,i} [\mu - f^+ - \xi] + \sigma^{-1} \mu_{,i} \\
&= \sigma^{-1} \left[\mu_{,i} - \sigma_{,i} z\right] \\
z_{,ij} &= -\sigma^{-2} \sigma_{,j} \left[\mu_{,i} - \sigma_{,i} z\right] + \sigma^{-1} \left[\mu_{,ij} - \sigma_{,ij} z - \sigma_{,i} z_{,j}\right] \\
&= \sigma^{-1} \left[\mu_{,ij} - \sigma_{,ij} z - \sigma_{,i} z_{,j} - \sigma_{,j} z_{,i}\right]
\end{aligned}$$

Now we differentiate with respect to data and hypers:

$$\begin{aligned}
\dot{z} &= -\sigma^{-2} \dot{\sigma} [\mu - f^+ - \xi] + \sigma^{-1} [\dot{\mu} - \dot{f}^+ - \dot{\xi}] \\
&= \sigma^{-1} [\dot{\mu} - \dot{f}^+ - \dot{\xi} - \dot{\sigma} z] \\
\dot{z}_{,i} &= \sigma^{-1} [\dot{\mu}_{,i} - \dot{\sigma}_{,i} z - \dot{\sigma} z_{,i} - \sigma_{,i} \dot{z}]
\end{aligned}$$

# F Differentiating $\sigma$

The predictive variance is

$$\sigma^2 = k_{xx} - k_{xX} K_{XX}^{-1} k_{Xx}.$$

Differentiating the predictive variance twice in space — assuming $k_{xx}$ is independent of $x$ by stationarity — gives us

$$\begin{aligned}
2\sigma\sigma_{,i} &= -2k_{xX,i} K_{XX}^{-1} k_{Xx} = -2k_{xX,i} d \\
2\sigma_{,i}\sigma_{,j} + 2\sigma\sigma_{,ij} &= -2k_{xX,ij} K_{XX}^{-1} k_{Xx} - 2k_{xX,i} K_{XX}^{-1} k_{Xx,j} \\
&= -2k_{xX,ij} d - 2k_{xX,i} w^{(j)}
\end{aligned}$$

Rearranging to get spatial derivatives of $\sigma$ on their own gives us

$$\begin{aligned}\sigma_{,i} &= -\sigma^{-1} k_{xX,i} d \\ \sigma_{,ij} &= -\sigma^{-1}\left[k_{xX,ij} d + k_{xX,i} w^{(j)} + \sigma_{,i}\sigma_{,j}\right].\end{aligned}$$

Differentiating with respect to data (and locations) and kernel hypers requires more work. First, note that

$$\begin{aligned}2\sigma\dot{\sigma} &= \dot{k}_{xx} - 2\dot{k}_{xX} K_{XX}^{-1} k_{Xx} + k_{xX} K_{XX}^{-1} \dot{K}_{XX} K_{XX}^{-1} k_{Xx} \\ &= \dot{k}_{xx} - 2\dot{k}_{xX} d + d^T \dot{K}_{XX} d\end{aligned}$$

Now, differentiating $\sigma^{-1}$ with respect to data and hypers gives

$$\begin{aligned}\dot{\sigma}_{,i} &= \sigma^{-2}\dot{\sigma} k_{xX,i} K_{XX}^{-1} k_{Xx} - \sigma^{-1}\left[\dot{k}_{xX,i} K_{XX}^{-1} k_{Xx} + k_{xX,i} K_{XX}^{-1} \dot{k}_{Xx} - k_{xX} K_{XX}^{-1} \dot{K}_{XX} K_{XX}^{-1} k_{Xx}\right] \\ &= -\sigma^{-1}\left[\dot{\sigma}\sigma_{,i} + \dot{k}_{xX,i} d + (w^{(i)})^T \dot{k}_{Xx} - d^T \dot{K}_{XX} d\right]\end{aligned}$$

# G Differentiating $\mu$

Let $K_{XX}$ denote the kernel matrix, and $k_{Xx}$ the column vector of kernel evaluations at $x$. The posterior mean function for the GP (assuming a zero-mean prior) is

$$\mu = k_{xX} c$$

where $K_{XX} c = y$. Note that $c$ does not depend on $x$, but it does depend on the data and hyperparameters.

Differentiating in space is straightforward, as we only invoke the kernel derivatives:

$$\begin{aligned}\mu_{,i} &= k_{xX,i} c \\ \mu_{,ij} &= k_{xX,ij} c\end{aligned}$$

Differentiating in the data and hyperparameters requires that we also differentiate through a matrix solve:

$$\dot{\mu} = \dot{k}_{xX} K_{XX}^{-1} y + k_{xX} K_{XX}^{-1} \dot{y} - k_{xX} K_{XX}^{-1} \dot{K}_{XX} K_{XX}^{-1} y.$$

Defining $d = K_{XX}^{-1} k_{Xx}$, we have

$$\dot{\mu} = \dot{k}_{xX} c + d^T (\dot{y} - \dot{K}_{XX} c).$$

Now differentiating in space and defining $K_{XX}^{-1} k_{Xx,i}$ as $w^{(i)}$, we have

$$\dot{\mu}_{,i} = \dot{k}_{xX,i} c + (w^{(i)})^T (\dot{y} - \dot{K}_{XX} c).$$

# H Differentiating kernels

We assume the kernel has the form $k(x, y) = \psi(\rho)$ where $\rho = \|r\|$ and $r = x - y$. Recall that

$$\begin{aligned}\rho &= \sqrt{r_k r_k} \\ \rho_{,i} &= \rho^{-1} r_k r_{k,i} = \rho^{-1} r_i \\ \rho_{,ij} &= \rho^{-1}\delta_{ij} - \rho^{-2} r_i \rho_{,j} \\ &= \rho^{-1}\left[\delta_{ij} - \rho^{-2} r_i r_j\right]\end{aligned}$$

Applying this together with the chain rule yields

$$\begin{aligned}k &= \psi(\rho) \\ k_{,i} &= \psi'(\rho)\rho_{,i} = \psi'(\rho)\rho^{-1} r_i \\ k_{,ij} &= \psi''(\rho)\rho_{,i}\rho_{,j} + \psi'(\rho)\rho_{,ij} \\ &= \left[\psi''(\rho) - \rho^{-1}\psi'(\rho)\right]\rho^{-2} r_i r_j + \rho^{-1}\psi'(\rho)\delta_{ij}.\end{aligned}$$

# NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

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

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

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

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: Much of the work in the space of non-myopic Bayesian optimization is about solving intractable integrals. We don't mention societal impact since the work is about developing an approach that mitigates issues associated with the intractability of the underlying problem.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [NA]

    Justification: This paper does not use existing assets.

    Guidelines:

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

13. **New Assets**

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

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects

Guidelines:

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