Optimization as Estimation with Gaussian Processes in Bandit Settings

Zi Wang
MIT CSAIL

Bolei Zhou
MIT CSAIL

Stefanie Jegelka
MIT CSAIL

Abstract

Recently, there has been rising interest in Bayesian optimization – the optimization of an unknown function with assumptions usually expressed by a Gaussian Process (GP) prior. We study an optimization strategy that directly uses an estimate of the argmax of the function. This strategy offers both practical and theoretical advantages: no tradeoff parameter needs to be selected, and, moreover, we establish close connections to the popular GP-UCB and GP-PI strategies. Our approach can be understood as automatically and adaptively trading off exploration and exploitation in GP-UCB and GP-PI. We illustrate the effects of this adaptive tuning via bounds on the regret as well as an extensive empirical evaluation on robotics and vision tasks, demonstrating the robustness of this strategy for a range of performance criteria.

1 Introduction

The optimization of an unknown function that is expensive to evaluate is an important problem in many areas of science and engineering. Bayesian optimization uses probabilistic methods to address this problem. In particular, an increasingly popular direction has been to model smoothness assumptions on the function via a Gaussian Process (GP). The Bayesian approach provides a posterior distribution of the unknown function, and thereby uncertainty estimates that help decide where to evaluate the function next, in search of a maximum. Recent successful applications of this *Bayesian optimization* framework include the tuning of hyperparameters for complex models and algorithms in machine learning, robotics, and computer vision [3, 6, 16, 19, 29, 33].

Despite progress on theory and applications of Bayesian

optimization methods, the practitioner continues to face many options: there is a menu of algorithms, and their relations and tradeoffs are only partially understood. Typically, the points where the function is evaluated are selected sequentially; and the choice of the next point is based on observed function values at the previous points. Popular algorithms vary in their strategies to pick the next point: they select the point that maximizes the *probability of improvement* (GP-PI) [17]; the *expected improvement* (GP-EI) [22]; or an *upper confidence bound* (GP-UCB) [30] on the maximum function value. Another alternative is *entropy search* (ES) [12], which aims to minimize the uncertainty about the location of the optimum of the function. Each algorithm reduces the black-box function optimization problem to a series of optimization problems of known *acquisition functions*.

The motivations and analyses (if available) differ too: objectives include *cumulative regret*, where every evaluation results in a reward or cost and the average of all function evaluations is compared to the maximum value of the function; *simple regret* that takes into account only the best value found so far [4]; the performance under a fixed finite budget [11]; or the uncertainty about the location of the function maximizer [12]. Here, we focus on the established objectives of cumulative regret in bandit games.

Notably, many of the above algorithms involve tuning a parameter to trade off exploration and exploitation, and this can pose difficulties in practice [12]. Theoretical analyses help in finding good parameter settings, but may be conservative in practice [30]. Computing the *acquisition function* and optimizing it to find the next point can be computationally very costly too. For example, the computation to decide which next point to evaluate for entropy search methods tends to be very expensive, while GP-PI, GP-EI and GP-UCB are much cheaper.

In this paper, we study an intuitive strategy that offers a compromise between a number of these approaches and, at the same time, establishes connections between them that help understand when theoretical results can be transferred. Our strategy uses the Gaussian Process to obtain an estimate of the argument that maximizes the unknown function f. The next point to evaluate is determined by this estimate. This point, it turns out, is not necessarily the same as the

the argument with the highest upper confidence bound.

This strategy has both practical and theoretical advantages. On the theoretical side, we show connections to the popular GP-UCB and GP-PI strategies, implying an intuitive and provably correct way of setting the parameters in those important methods. Moreover, we establish bounds on the regret of our estimation strategy. From a practical viewpoint, our strategy obviates any costly parameter tuning. In fact, we show that it corresponds to automatically and *adaptively* tuning the parameters of GP-UCB and GP-PI. Our empirical evaluation includes problems from non-convex optimization, robotics, and computer vision. The experiments show that our strategy performs similarly to or even better than the best competitors in terms of cumulative regret. Although not designed to minimize simple regret directly, in practice our method also works well as measured by simple regret, or by the number of steps to reach a fixed regret value. Together, these results suggest that our strategy is easy to use and empirically performs well across a spectrum of settings.

Related work. The practical benefits of Bayesian optimization have been shown in a number of applications [3, 6, 19, 29, 33, 34]. Different Bayesian optimization algorithms differ in the selection criteria of the next point to evaluate, i.e., the acquisition function. Popular criteria include the expected improvement (GP-EI) [22], the probability of improving over a given threshold (GP-PI) [17], and GP-UCB [30], which is motivated by upper confidence bounds for multi-armed bandit problems [2, 1]. GP-EI, GP-PI and GP-UCB have a parameter to select, and the latter two are known to be sensitive to this choice. Entropy search (ES) [12] and the related predictive entropy search (PES) [13] do not aim to minimize regret directly, but to maximize the amount of information gained about the optimal point. High-dimensional settings were considered in [7, 34]. Extensive empirical comparisons include [6, 29, 20]. Theoretical bounds on different forms of regret were established for GP-UCB [30] and GP-EI [5]. Other theoretical studies focus on simple regret [4, 11] or finite budgets [11]. In this work, in contrast, we are motivated by practical considerations.

1.1 Background and Notation

Let $f(\cdot) \sim GP(0, k)$ be an unknown function we aim to optimize over a candidate set $\mathfrak{X}$. At time step t, we select point $\boldsymbol{x}_t$ and observe a possibly noisy function evaluation $y_t = f(\boldsymbol{x}_t) + \epsilon_t$, where ϵ_t are i.i.d. Gaussian noise $\mathcal{N}(0, \sigma^2)$. Given the observations $\mathfrak{D}_t = \{(\boldsymbol{x}_\tau, y_\tau)\}_{\tau=1}^t$ up to time t, we obtain the posterior mean and covariance of the function via the kernel matrix $\boldsymbol{K}_t = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathfrak{D}_t}$ and $\boldsymbol{k}_t(x) = [k(\boldsymbol{x}_i, \boldsymbol{x})]_{\boldsymbol{x}_i \in \mathfrak{D}_t}$ [24]: $\mu_t(\boldsymbol{x}) = \boldsymbol{k}_t(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{K}_t + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{y}_t$, and $k_t(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_t(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{K}_t + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{k}_t(\boldsymbol{x}')$. The posterior variance is given by $\sigma_t^2(\boldsymbol{x}) = k_t(\boldsymbol{x}, \boldsymbol{x})$. Furthermore, we denote by $Q(\cdot)$ the tail probability of the standard normal distribution $\phi(\cdot)$, and by $\Phi(\cdot)$ its cumulative probability.

The Bayesian Optimization setting corresponds to a bandit game where, in each round t, the player chooses a point $\boldsymbol{x}_t$ and then observes $y_t = f(\boldsymbol{x}_t) + \epsilon_t$. The *regret* for round t is defined as $\tilde{r}_t = \max_{\boldsymbol{x} \in \mathfrak{X}} f(\boldsymbol{x}) - f(\boldsymbol{x}_t)$. The *simple regret* for any T rounds is $r_T = \min_{t \in [1, T]} \tilde{r}_t$, and the (average) cumulative regret is $R_T = \frac{1}{T} \sum_{t=1}^T \tilde{r}_t$.

1.2 Existing methods for GP optimization

We focus on the following three approaches for comparison, since they are most widely used in bandit settings.

GP-UCB. Srinivas et al. [30] provide a detailed analysis for using upper confidence bounds [2] with GP bandits. They propose the strategy $\boldsymbol{x}_t = \arg\max_{\boldsymbol{x} \in \mathfrak{X}} \mu_{t-1}(\boldsymbol{x}) + \lambda_t \sigma_{t-1}(\boldsymbol{x})$ where $\lambda_t = (2 \log(|\mathfrak{X}| \pi^2 t^2 / (6\delta)))^{\frac{1}{2}}$ for finite $\mathfrak{X}$. Their regret bound holds with probability $1 - \delta$.

GP-EI. The GP-EI strategy [22] selects the point maximizing the expected improvement over a pre-specified threshold θ_t [22]. For GPs, this improvement is given in closed form as $\mathrm{EI}(x) = \mathbb{E}[(f(x) - \theta_t)_+] = [\phi(\boldsymbol{\gamma}(\boldsymbol{x})) - \boldsymbol{\gamma}(\boldsymbol{x}) Q(\boldsymbol{\gamma}(\boldsymbol{x}))] \sigma_{t-1}(\boldsymbol{x})$, where $\boldsymbol{\gamma}(\boldsymbol{x}) = \frac{\theta_t - \mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$. A popular choice for the threshold is $\theta_t = \max_{\tau \in [1, t-1]} y_\tau$.

GP-PI. The third strategy maximizes the probability $\mathrm{PI}(x) = \Pr[f(x) > \theta_t] = 1 - \Phi(\boldsymbol{\gamma}(\boldsymbol{x}))$ of improving over a threshold θ_t [17], i.e., $\boldsymbol{x}_t = \arg\min_{\boldsymbol{x} \in \mathfrak{X}} \boldsymbol{\gamma}(\boldsymbol{x})$. GP-PI is sensitive to the choice of θ_t: as we will see in Section 2, θ_t trades off exploration and exploitation, and setting θ_t too low (e.g., $\theta_t = \max_{\tau \in [1, t-1]} y_\tau$) can result in getting stuck at a fairly suboptimal point. A popular choice in practice is $\theta_t = \max_{\tau \in [1, t-1]} y_\tau + \epsilon$, for a chosen constant ϵ.

2 Optimization as estimation

In this work, we study an alternative criterion that provides an easy-to-use and tuning-free approach: we use the GP to estimate the $\arg\max$ of f. In Section 2.1, we will see how, as a side effect, this criterion establishes connections between the above criteria. Our strategy eventually leads to tighter bounds than GP-UCB as shown in Section 3.

Consider the posterior probability (in round t) that a fixed $\boldsymbol{x} \in \mathfrak{X}$ is an $\arg\max$ of f. We call this event $M_{\boldsymbol{x}}$ and, for notational simplicity, omit the subscripts $t - 1$ here. The event $M_{\boldsymbol{x}}$ is equivalent to the event that for all $\boldsymbol{x}' \in \mathfrak{X}$, we have $v(\boldsymbol{x}') := f(\boldsymbol{x}') - f(\boldsymbol{x}) \leq 0$. The difference $v(\boldsymbol{x}')$ between two Gaussian variables is Gaussian: $v(\boldsymbol{x}') \sim \mathcal{N}(\mu(\boldsymbol{x}') - \mu(\boldsymbol{x}), \sigma(\boldsymbol{x})^2 + \sigma(\boldsymbol{x}')^2 - 2k(\boldsymbol{x}, \boldsymbol{x}')^2)$. The covariance for any $\boldsymbol{x}', \boldsymbol{x}'' \in \mathfrak{X}$ is $\mathrm{Cov}(v(\boldsymbol{x}'), v(\boldsymbol{x}'')) = \sigma(\boldsymbol{x})^2 + k(\boldsymbol{x}', \boldsymbol{x}'')^2 - k(\boldsymbol{x}, \boldsymbol{x}')^2 - k(\boldsymbol{x}, \boldsymbol{x}'')^2$.

The random variables $\{v(\boldsymbol{x}')\}_{\boldsymbol{x}' \in \mathfrak{X}}$ determine the cumulative probability

$$\Pr[M_{\boldsymbol{x}}|\mathfrak{D}] = \Pr[\forall \boldsymbol{x}' \in \mathfrak{X}, v(\boldsymbol{x}') \leq 0|\mathfrak{D}]. \quad (1)$$

This probability may be specified via limits as e.g. in [12, App.A]. Moreover, due to the assumed smoothness of f, it is reasonable to work with a discrete approximation and restrict the set of candidate points to be finite for now (we discuss discretization further in Section 5). So the quantity in Eqn. (1) is well-defined. Since computing $\Pr[M_{\boldsymbol{x}}|\mathfrak{D}]$ for large $|\mathfrak{X}|$ can be costly, we use a "mean-field" approach and approximate $\{f(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathfrak{X}}$ by independent Gaussian random variables with means $\mu(\boldsymbol{x})$ and variances $\sigma(\boldsymbol{x})^2$ for all $\boldsymbol{x} \in \mathfrak{X}$. Given be the maximum value m of f, the probability of the event $M_{\boldsymbol{x}}|m, \mathfrak{D}$ amounts to

$$\Pr[M_{\boldsymbol{x}}|m, \mathfrak{D}] \approx Q\Big(\frac{m - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\Big) \prod_{\boldsymbol{x}' \neq \boldsymbol{x}} \Phi\Big(\frac{m - \mu(\boldsymbol{x}')}{\sigma(\boldsymbol{x}')}\Big).$$

Our estimation strategy (EST) chooses to evaluate $\arg\max_{\boldsymbol{x} \in \mathfrak{X}} \Pr[M_{\boldsymbol{x}}|\hat{m}, \mathfrak{D}]$ next, which is the function input that is most likely to achieve the highest function value.

Of course, the function maximum m may be unknown. In this case, we use a plug-in estimate via the posterior expectation of $Y = \max_{\boldsymbol{x} \in \mathfrak{X}} f(\boldsymbol{x})$ given $\mathfrak{D}$ [26]:

$$\hat{m} = \mathbb{E}[Y|\mathfrak{D}] \quad (2)$$

$$= \int_0^\infty \Pr[Y > y|\mathfrak{D}] - \Pr[Y < -y|\mathfrak{D}]\,\mathrm{d}y. \quad (3)$$

If the noise in the observations is negligible, we can simplify Eqn. (3) to be

$$\hat{m} = m_0 + \int_{m_0}^\infty 1 - \prod_{\boldsymbol{x} \in \mathfrak{X}} \Phi\Big(\frac{w - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\Big)\,\mathrm{d}w \quad (4)$$

where $m_0 = \max_{\tau \in [1, t-1]} y_\tau$ is the current observed maximum value. Under conditions specified in Section 3, our approximation with the independence assumption makes $\hat{m}$ an upper bound on m, which, as we will see, conservatively emphasizes exploration a bit more. Other ways of setting $\hat{m}$ are discussed in Section 5.

2.1 Connections to GP-UCB and GP-PI

Next, we relate our strategy to GP-PI and GP-UCB: EST turns out to be equivalent to adaptively tuning θ_t in GP-PI and λ_t in GP-UCB. This observation reveals unifying connections between GP-PI and GP-UCB and, in Section 3, yields regret bounds for GP-PI with a certain choice of θ_t. Lemma 2.1 characterizes the connection to GP-UCB:

Lemma 2.1. *In any round t, the point selected by EST is the same as the point selected by a variant of GP-UCB with $\lambda_t = \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m}_t - \mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$. Conversely, the candidate selected by GP-UCB is the same as the candidate selected by a variant of EST with $\hat{m}_t = \max_{\boldsymbol{x} \in \mathfrak{X}} \mu_{t-1}(\boldsymbol{x}) + \lambda_t \sigma_{t-1}(\boldsymbol{x})$.*

Proof. We omit the subscripts t for simplicity. Let $\boldsymbol{a}$ be the point selected by GP-UCB, and $\boldsymbol{b}$ selected by EST. Without loss of generality, we assume $\boldsymbol{a}$ and $\boldsymbol{b}$ are unique. With $\lambda = \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$, GP-UCB chooses to evaluate

$$\boldsymbol{a} = \arg\max_{\boldsymbol{x} \in \mathfrak{X}} \mu(\boldsymbol{x}) + \lambda\sigma(\boldsymbol{x}) = \arg\min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}.$$

This is because

$$\hat{m} = \max_{\boldsymbol{x} \in \mathfrak{X}} \mu(\boldsymbol{x}) + \lambda\sigma(\boldsymbol{x}) = \mu(\boldsymbol{a}) + \lambda\sigma(\boldsymbol{a}).$$

By definition of $\boldsymbol{b}$, for all $\boldsymbol{x} \in \mathfrak{X}$, we have

$$\frac{\Pr[M_{\boldsymbol{b}}|\hat{m}, \mathfrak{D}]}{\Pr[M_{\boldsymbol{x}}|\hat{m}, \mathfrak{D}]} \approx \frac{Q(\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})})\Phi(\frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})})}{Q(\frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})})\Phi(\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})})} \geq 1.$$

The inequality holds if and only if $\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})} \leq \frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$ for all $\boldsymbol{x} \in \mathfrak{X}$, including $\boldsymbol{a}$, and hence

$$\frac{\hat{m} - \mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})} \leq \frac{\hat{m} - \mu(\boldsymbol{a})}{\sigma(\boldsymbol{a})} = \lambda = \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})},$$

which, with uniqueness, implies that $\boldsymbol{a} = \boldsymbol{b}$ and GP-UCB and EST select the same point.

The other direction of the proof is similar and can be found in the supplement. $\square$

Proposition 2.2. *GP-PI is equivalent to EST when setting $\theta_t = \hat{m}_t$ in GP-PI.*

As a corollary of Lemma 2.1 and Proposition 2.2, we obtain a correspondence between GP-PI and GP-UCB.

Corollary 2.3. *GP-UCB is equivalent to GP-PI if λ_t is set to $\min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\theta_t - \mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$, and GP-PI corresponds to GP-UCB if $\theta_t = \max_{\boldsymbol{x} \in \mathfrak{X}} \mu_{t-1}(\boldsymbol{x}) + \lambda_t \sigma_{t-1}(\boldsymbol{x})$.*

Algorithm 1 GP-UCB/PI/EST

1: $t \leftarrow 1; \mathfrak{D}_0 \leftarrow \emptyset$
2: **while** stopping criterion not reached **do**
3: $\quad \mu_{t-1}, \Sigma_{t-1} \leftarrow$ GP-predict$(\mathfrak{X}|\mathfrak{D}_{t-1})$
4: $\quad \hat{m}_t = \begin{cases} \max_{\boldsymbol{x} \in \mathfrak{X}} \mu_{t-1}(\boldsymbol{x}) + \lambda_t \sigma_{t-1}(\boldsymbol{x}) & \text{GP-UCB} \\ \max_{1 \leq \tau < t} y_\tau + \epsilon & \text{GP-PI} \\ \mathbb{E}[Y|\mathfrak{D}_{t-1}] \text{ (see Eqn. (3)/Eqn. (4))} & \text{EST} \end{cases}$
5: $\quad \boldsymbol{x}_t \leftarrow \arg\min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m}_t - \mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$
6: $\quad y_t \leftarrow f(\boldsymbol{x}_t) + \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \sigma^2)$
7: $\quad \mathfrak{D}_t \leftarrow \{\boldsymbol{x}_\tau, y_\tau\}_{\tau=1}^t$
8: $\quad t \leftarrow t + 1$
9: **end while**

Proposition 2.2 suggests that we do not need to calculate the probability $\Pr[M_{\boldsymbol{x}}|\hat{m}_t, \mathfrak{D}_{t-1}]$ directly when implementing EST. Instead, we can reduce EST to GP-PI with an automatically tuned target value θ_t. Algorithm 1 compares the pseudocode for all three methods. We use

"GP-predict" to denote the update for the posterior mean and covariance function for the GP as described in Section 1.1. GP-UCB/PI/EST all share the same idea of reaching a target value ($\hat{m}_t$ in this case), and thereby trading off exploration and exploitation. GP-UCB in [30] can be interpreted as setting the target value to be a loose upper bound $\max_{\boldsymbol{x} \in \mathfrak{X}} \mu_{t-1}(\boldsymbol{x}) + \lambda_t \sigma_{t-1}(\boldsymbol{x})$ with $\lambda_t = (2 \log(|\mathfrak{X}|\pi^2 t^2/6\delta))^{\frac{1}{2}}$, as a result of applying the union bound over $\mathfrak{X}$[1]. GP-PI applies a fixed upwards shift of ϵ over the current maximum observation $\max_{\tau \in [1,t-1]} y_\tau$. In both cases, the exploration-exploitation tradeoff depends on the parameter to be set. EST implicitly and automatically balances the two by estimating the maximum. Viewed as GP-UCB or GP-PI, it automatically sets the respective parameter.

Note that this change by EST is not only intuitively reasonable, but it also leads to vanishing regret, as will become evident in the next section.

3 Regret Bounds

In this section, we analyze the regret of EST. We first show a bound on the cumulative regret both in expectation and with high probability, with the assumption that our estimation $\hat{m}_t$ is always an upper bound on the maximum of the function. Then we interpret how this assumption is satisfied via Eqn. (3) and Eqn. (4) under the condition specified in Corollary 3.5.

Theorem 3.1. *We assume $\hat{m}_t \geq \max_{\boldsymbol{x} \in \mathfrak{X}} f(\boldsymbol{x}), \forall t \in [1, T]$, and restrict $k(\boldsymbol{x}, \boldsymbol{x}') \leq 1$. Let σ^2 be the variance of the Gaussian noise in the observation, γ_T the maximum information gain of the selected points, $C = 2/\log(1 + \sigma^{-2})$, and $t^* = \arg\max_t \nu_t$ where $\nu_t \triangleq \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m}_t - \mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$. The cumulative expected regret satisfies $\sum_{t=1}^T \mathbb{E}[\tilde{r}_t | \mathfrak{D}_{t-1}] \leq \nu_{t^*} \sqrt{CT\gamma_T}$. With probability at least $1 - \delta$, it holds that $\sum_{t=1}^T \tilde{r}_t \leq (\nu_{t^*} + \zeta_T) \sqrt{CT\gamma_T}$, with $\zeta_T = (2 \log(\frac{T}{2\delta}))^{\frac{1}{2}}$.*

The information gain γ_T after T rounds is the maximum mutual information that can be gained about f from T measurements: $\gamma_T = \max_{A \subseteq \mathfrak{X}, |A| \leq T} I(\boldsymbol{y}_A, \boldsymbol{f}_A) = \max_{A \subseteq \mathfrak{X}, |A| \leq T} \frac{1}{2} \log \det(\boldsymbol{I} + \sigma^{-2} \boldsymbol{K}_A)$. For the Gaussian kernel, $\gamma_T = O((\log T)^{d+1})$, and for the Matérn kernel, $\gamma_T = O(T^{d(d+1)/(2\xi+d(d+1))} \log T)$ where d is the dimension and ξ is the roughness parameter of the kernel [30, Theorem 5].

The proof of Theorem 3.1 follows [30] and relies on the following lemmas which are proved in the supplement.

Lemma 3.2. *Pick $\delta \in (0, 1)$ and set $\zeta_t = (2 \log(\frac{\pi_t}{2\delta}))^{\frac{1}{2}}$, where $\sum_{t=1}^T \pi_t^{-1} \leq 1$, $\pi_t > 0$. Then, for EST, it holds that $\Pr[\mu_{t-1}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t) \leq \zeta_t \sigma_{t-1}(\boldsymbol{x}_t)] \geq 1 - \delta$, for all $t \in [1, T]$.*

Lemma 3.2 is similar to but not exactly the same as [30, Appendix A.1]: while they use a union bound over all of $\mathfrak{X}$, here, we only need a union bound over the actually evaluated points $\{\boldsymbol{x}_t\}_{t=1}^T$. This difference is due to the different selection strategies.

Lemma 3.3. *If $\mu_{t-1}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t) \leq \zeta_t \sigma_{t-1}(\boldsymbol{x}_t)$, the regret at time step t is upper bounded as $\tilde{r}_t \leq (\nu_t + \zeta_t)\sigma_{t-1}(\boldsymbol{x}_t)$, where $\nu_t \triangleq \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m}_t - \mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$, and $\hat{m}_t \geq \max_{\boldsymbol{x} \in \mathfrak{X}} f(\boldsymbol{x}), \forall t \in [1, T]$.*

The proof of Theorem 3.1 now follows from Lemmas 3.2, 3.3, and Lemma 5.3 in [30].

Proof. (Thm. 3.1) The expected regret of round t is $\mathbb{E}[\tilde{r}_t | \mathfrak{D}_{t-1}] \leq \hat{m}_t - \mu_{t-1}(\boldsymbol{x}_t) = \nu_t \sigma_{t-1}(\boldsymbol{x}_t)$. Using $t^* = \arg\max_t \nu_t$, we obtain that $\sum_{t=1}^T \mathbb{E}[\tilde{r}_t | \mathfrak{D}_{t-1}] \leq \nu_{t^*} \sum_{t=1}^T \sigma_{t-1}(\boldsymbol{x}_t)$.

To bound the sum of variances, we first use that $(1 + a)^x \leq 1 + ax$ for $0 \leq x \leq 1$ and the assumption $\sigma_{t-1}(\boldsymbol{x}_t) \leq k(\boldsymbol{x}_t, \boldsymbol{x}_t) \leq 1$ to obtain $\sigma_{t-1}^2(\boldsymbol{x}_t) \leq \frac{\log(1 + \sigma^{-2}\sigma_{t-1}^2(\boldsymbol{x}_t))}{\log(1 + \sigma^{-2})}$. Lemma 5.3 in [30] now implies that $\sum_{t=1}^T \sigma_{t-1}^2(\boldsymbol{x}_t) \leq \frac{2}{\log(1+\sigma^{-2})} I(\boldsymbol{y}_T; \boldsymbol{f}_T) \leq \frac{2}{\log(1+\sigma^{-2})} \gamma_T$. The Cauchy-Schwarz inequality leads to $\sum_{t=1}^T \sigma_{t-1}(\boldsymbol{x}_t) \leq \sqrt{T \sum_{t=1}^T \sigma_{t-1}^2(\boldsymbol{x}_t)} \leq \sqrt{\frac{2T\gamma_T}{\log(1+\sigma^{-2})}}$. Together, we have the final regret bound

$$\sum_{t=1}^T \mathbb{E}[\tilde{r}_t | \mathfrak{D}_{t-1}] \leq \nu_{t^*} \sqrt{\frac{2T}{\log(1 + \sigma^{-2})} \gamma_T}.$$

Next we show a high probability bound. The condition of Lemma 3.3 holds with high probability because of Lemma 3.2. Thus with probability at least $1 - \delta$, the regret for round t is bounded as follows,

$$\tilde{r}_t \leq (\nu_t + \zeta_t)\sigma_{t-1}(\boldsymbol{x}_t) \leq (\nu_{t^*} + \zeta_T)\sigma_{t-1}(\boldsymbol{x}_t),$$

where $\zeta_t = 2 \log(\frac{\pi_t}{2\delta})$, $\pi_t = \frac{\pi^2 t^2}{6}$, and $t^* = \arg\max_t \nu_t$. Therefore, with probability at least $1 - \delta$,

$$\sum_{t=1}^T \tilde{r}_t \leq (\nu_{t^*} + \zeta_T)\sqrt{\frac{2T\gamma_T}{\log(1 + \sigma^{-2})}}. \quad \square$$

Next we show that if we estimate $\hat{m}_t$ as described in Section 2 by assuming all the $\{f(\boldsymbol{x})\}_{\boldsymbol{x} \in \mathfrak{X}}$ are independent conditioned on the current sampled data $\mathfrak{D}_t$, $\hat{m}_t$ can be guaranteed to be an upper bound on the function maximum m given $k_t(\boldsymbol{x}, \boldsymbol{x}') \geq 0, \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathfrak{X}$.

---

[1] Since $\Pr[|f(\boldsymbol{x}) - \mu(\boldsymbol{x})| > \lambda_t \sigma(\boldsymbol{x})] \leq e^{\frac{-\lambda_t^2}{2}}$, applying the union bound results in $\Pr[|f(\boldsymbol{x}) - \mu(\boldsymbol{x})| > \lambda_t \sigma(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathfrak{X}] \leq |\mathfrak{X}| e^{\frac{-\lambda_t^2}{2}}$. This means $f(\boldsymbol{x}) \leq \max_{\boldsymbol{x} \in \mathfrak{X}} \mu(\boldsymbol{x}) + \lambda_t \sigma(\boldsymbol{x})$ with probability at least $1 - |\mathfrak{X}| e^{\frac{-\lambda_t^2}{2}}$ [30, Lemma 5.1].

Lemma 3.4 (Slepian's Comparison Lemma [28, 21]). *Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ be two multivariate Gaussian random vectors with the same mean and variance, such that $\mathbb{E}[\boldsymbol{v}_i \boldsymbol{v}_j] \leq \mathbb{E}[\boldsymbol{u}_i \boldsymbol{u}_j], \forall i, j$. Then $\mathbb{E}[\sup_{i \in [1,n]} \boldsymbol{v}_i] \geq \mathbb{E}[\sup_{i \in [1,n]} \boldsymbol{u}_i]$.*

Slepian's Lemma implies a relation between our approximation $\hat{m}_t$ and m.

Corollary 3.5. *Assume $g \sim GP(\mu, k)$ has posterior mean $\mu_t(\boldsymbol{x})$ and posterior covariance $k_t(\boldsymbol{x}, \boldsymbol{x}') \geq 0, \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathfrak{X}$ conditioned on $\mathfrak{D}_t$. Define a series of* independent *random variables $h(\boldsymbol{x})$ with equivalent mean $\mu_t(\boldsymbol{x})$ and posterior variance $k_t(\boldsymbol{x}, \boldsymbol{x}), \forall \boldsymbol{x} \in \mathfrak{X}$. Then, $\mathbb{E}[\sup_{\boldsymbol{x} \in \mathfrak{X}} h(\boldsymbol{x})] \geq \mathbb{E}[\sup_{\boldsymbol{x} \in \mathfrak{X}} g(\boldsymbol{x})]$.*

Proof. By independence, $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathfrak{X}$

$$0 = \mathbb{E}[h(\boldsymbol{x})h(\boldsymbol{x}')] - \mathbb{E}[h(\boldsymbol{x})]\mathbb{E}[h(\boldsymbol{x}')]$$
$$\leq E[g(\boldsymbol{x})g(\boldsymbol{x}')] - \mathbb{E}[g(\boldsymbol{x})]\mathbb{E}[g(\boldsymbol{x}')]$$

Hence, Slepian's Lemma implies that $\mathbb{E}[\sup_{\boldsymbol{x} \in \mathfrak{X}} h(\boldsymbol{x})] \geq \mathbb{E}[\sup_{\boldsymbol{x} \in \mathfrak{X}} g(\boldsymbol{x})]$. $\square$

Corollary 3.5 assumes that $k_t(\boldsymbol{x}, \boldsymbol{x}') \geq 0, \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathfrak{X}$. This depends on the choice of $\mathfrak{X}$ and k. Notice that $k_t(\boldsymbol{x}, \boldsymbol{x}') \geq 0$ is only a sufficient condition and, even if the assumption fails, $\hat{m}_t$ is often still an upper bound on m in practice (illustrations in the supplement).

In contrast, the results above are not necessarily true for any arbitrary θ_t in GP-PI, an important distinction between GP-PI and GP-EST.

Before evaluating the EST strategy empirically, we make a few important observations. First, EST does not require manually setting a parameter that trades off exploration and exploitation. Instead, it corresponds to automatically, adaptively setting the tradeoff parameters λ_t in GP-UCB and θ_t in GP-PI. For EST, this means that if the gap ν_t is large, then the method focuses more on exploration, and if "good" function values (i.e., close to $\hat{m}_t$) are observed, then exploitation increases. If we write EST as GP-PI, we see that by Eqn. (4), the estimated $\hat{m}_t$ always ensures $\theta_t > \max_{\tau \in [1,T]} y_\tau$, which is known to be advantageous in practice [20]. These analogies likewise suggest that $\theta_t = \max_{\tau \in [1,T]} y_\tau$ corresponds to a very small λ_t in GP-UCB, and results in very little exploration, offering an explanation for the known shortcomings of this θ_t.

4 Experiments

We test EST[2] in three domains: (1) synthetic black box functions; (2) initialization tuning for trajectory optimization; and (3) parameter tuning for image classification. We compare the following methods: EST with a

---

[2]The code is available at `https://github.com/zi-w/GP-EST`.

Table 1: Minimum Regret $r(t)$ and time to achieve this minimum for functions sampled from 1-D (top) and 2-D Gaussian Processes (bottom) with a limited budget of iterations. ESTa and ESTn achieve lower regret values ($r_{\min}$) faster than other methods ($T_{\min}$). Here, '‾' denotes the mean and '^' the median.

1-D GP, max 150 rounds						
	RAND	UCB	EI	PI	ESTa	ESTn
$\hat{T}_{min}$	79.5	53	8	7	26	23
$\hat{r}_{min}$	0.051	0.000	0.088	0.487	0.000	0.000
$\bar{T}_{min}$	78.4	50.9	9.77	8.32	26.1	21.9
$\bar{r}_{min}$	0.107	0.000	0.295	0.562	0.024	0.043
2-D GP, max 1000 rounds						
	RAND	UCB	EI	PI	ESTa	ESTn
$\hat{T}_{min}$	450.5	641.5	40.5	45	407.5	181
$\hat{r}_{min}$	0.640	0.090	1.035	1.290	0.000	0.000
$\bar{T}_{min}$	496.3	573.8	48.4	59.8	420.7	213.4
$\bar{r}_{min}$	0.671	0.108	0.976	1.26	0.021	0.085

Laplace approximation, i.e., approximating the integrand in Eqn. (4) by a truncated Gaussian (ESTa, details in supplement); EST with numerical integration to evaluate Eqn. (4) (ESTn); UCB; EI; PI; and random selection (RAND). We omit the 'GP-' prefix for simplicity.

For UCB, we follow [30] to set λ_t with $\delta = 0.01$. For PI, we use $\epsilon = 0.1$. These parameters are coarsely tuned via cross validation to ensure a small number of iterations for achieving low regret. Additional experimental results and details may be found in the supplement.

4.1 Synthetic Data

We sampled 200 functions from a 1-D GP and 100 functions from a 2-D GP with known priors (Matérn kernel and linear mean function). The maximum number of rounds was 150 for 1-D and 1000 for 2-D. The first samples were the same for all the methods to minimize randomness effects. Table 1 shows the lowest simple regret achieved ($r_{\min}$) and the number of rounds needed to reach it ($T_{\min}$). We measure the mean ($\bar{T}_{\min}$ and $\bar{r}_{\min}$) and the median ($\hat{T}_{\min}$ and $\hat{r}_{\min}$). Figure 1(a) illustrates the average simple regret and the standard deviation (scaled by $1/4$). While the progress of EI and PI quickly levels off, the other methods continue to reduce the regret, ESTn being the fastest. Moreover, the standard deviation of ESTa and ESTn is much lower than that of PI and EI. RAND is inferior both in terms of $T_{\min}$ and $r_{\min}$. UCB finds a good point but takes more than twice as many rounds as EST for doing so.

Figure 1(b) shows the cumulative regret R_T. As for the simple regret, we see that EI and PI focus too much on exploitation, stalling at a suboptimal point. EST converges to lower values, and faster than UCB. For additional intuition on cumulative regret, Figure 1(c) plots the cumulative
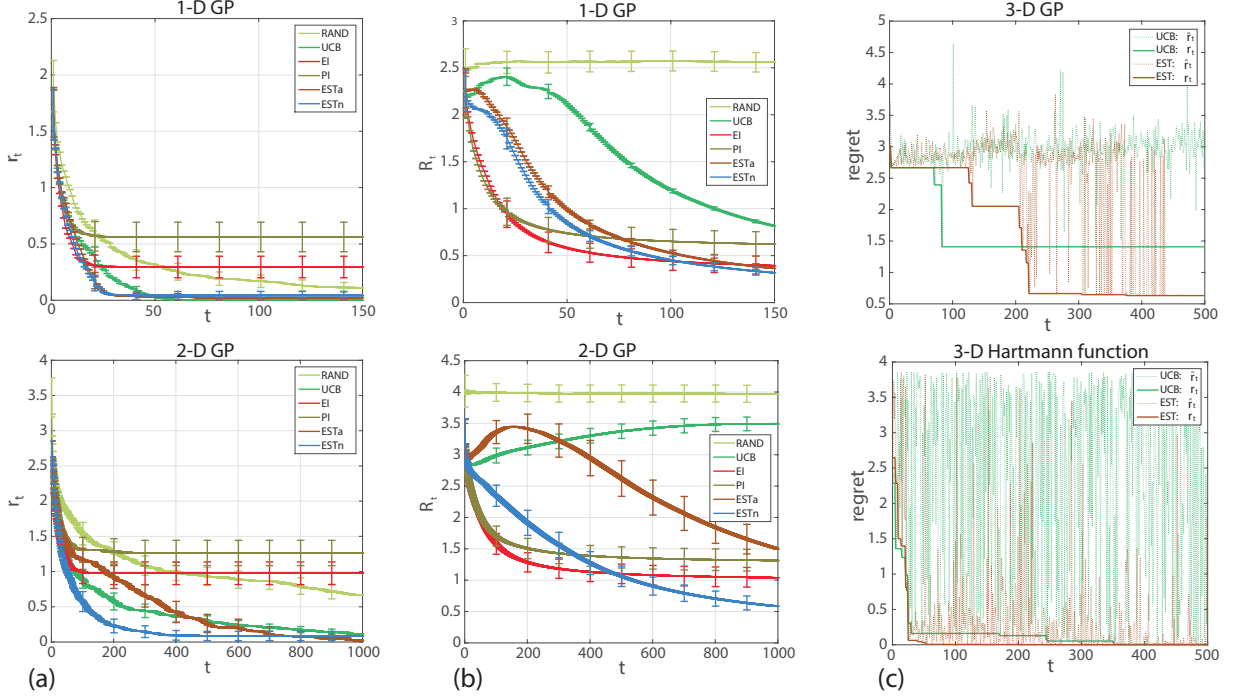
Figure 1: (a) Simple regrets for functions sampled from 1-D GP and 2-D GP over the number of rounds. ESTn quickly minimizes regret. (b) Cumulative regrets for functions sampled from 1-D GP and 2-D GP. ESTn lowers the regret more than PI and EI and faster than UCB. (c) Regrets $\tilde{r}_t$ and simple regrets r_t for a 3-D function sampled from GP and the 3-D Hartmann function. Here, EST is ESTa. For EST, the regret in each round is usually less than that of UCB, explaining UCB's higher cumulative regret.

regret $\tilde{r}_t$ and the simple regret r_t for UCB and EST for a function sampled from 3-D GP and a standard optimization test function (the 3-D Hartmann function). UCB tends to have higher cumulative regret than other methods because it keeps exploring drastically even after having reached the optimum of the function. This observation is in agreement with the experimental results in [30] (they improved UCB's performance by scaling λ_t down by a factor of 5), indicating that the scheme of setting λ_t in UCB is not always ideal in practice.

In summary, the results for synthetic functions suggest that throughout, compared to other methods, EST finds better function values within a smaller number of iterations.

4.2 Initialization Tuning for Trajectory Optimization

In online planning, robots must make a decision quickly, within a possibly unknown budget (humans can stop the "thinking period" of the robot any time, asking for a feasible and good decision to execute). We consider the problem of trajectory optimization, a non-convex optimization problem that is commonly solved via sequential quadratic programming (SQP) [27]. The employed solvers suffer from sub-optimal local optima, and, in real-world scenarios, even merely reaching a feasible solution can be challenging. Hence, we use Bayesian Optimization to tune the

Table 2: Lowest reward attained in 20 rounds on the airplane problem (illustrated in Figure 2). The results are averages over 8 settings.

	RAND	UCB	EI
mean	27.4429	29.2021	29.1832
std	3.3962	3.1247	3.1377
	PI	ESTa	ESTn
mean	28.0214	27.7587	29.2071
std	4.1489	4.2783	3.1171

initialization for trajectory optimization. In this setting, $\boldsymbol{x}$ is a trajectory initialization, and $f(\boldsymbol{x})$ the score of the solution returned by the solver after starting it at $\boldsymbol{x}$.

Our test example is the 2D airplane problem from [32], illustrated in Figure 2. We used 8 configurations of the starting point and fixed the target. Our candidate set $\mathfrak{X}$ of initializations is a grid of the first two dimensions of the midpoint state of the trajectory (we do not optimize over speed here). To solve the SQP, we used SNOPT [9]. Figure 2 shows the maximum rewards achieved up to round t (standard deviations scaled by 0.1), and Table 2 displays the final rewards. ESTn achieves rewards on par with the best competitors. Importantly, we observe that Bayesian optimization achieves much better results than the standard
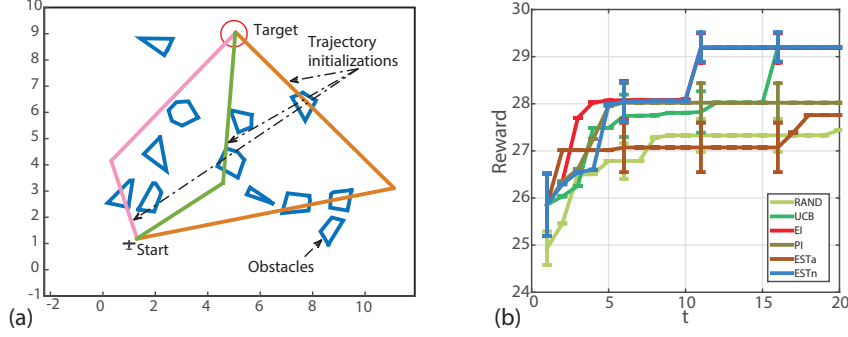
Figure 2: (a) Illustration of trajectory initializations. Trajectory initializations are passed to the non-convex optimization solver to ensure a solution with as few collisions with the obstacles as possible. (b) Maximum rewards up to round t. EI and ESTn perform relatively better than other methods.

random restarts, indicating a new successful application of Bayesian optimization.

4.3 Parameter Tuning for Image Classification

Our third set of experiments addresses Bayesian optimization for efficiently tuning parameters in visual classification tasks. Here, $\boldsymbol{x}$ is the model parameter and y the accuracy on the validation set. Our six image datasets are standard benchmarks for object classification (Caltech101 [8] and Caltech256 [10]), scene classification (Indoor67 [23] and SUN397 [35]), and action/event classification (Action40 [36] and Event8 [18]). The number of images per data set varies from 1,500 to 100,000. We use deep CNN features pre-trained on ImageNet [14], the state of the art on various visual classification tasks [25].

Our experimental setup follows [37]. The data is split into training, validation (20% of the original training set) and test set following the standard settings of the datasets. We train a linear SVM using the deep features, and tune its regularization parameter C via Bayesian optimization on the validation set. After obtaining the parameter recommended by each method, we train the classifier on the whole training set, and then evaluate on the test set.

Figure 3 shows the maximum achieved accuracy on the validation set during the iterations of Bayesian optimization on the six datasets. While all methods improve the classification accuracy, ESTa does so faster than other methods. Here too, PI and EI seem to explore too little. Table 3 displays the accuracy on the test set using the best parameter found by ESTa and ESTn, indicating that the parameter tuning via EST improved classification accuracy. For example, the tuning improves the accuracy on Action40 and SUN397 by 3-4% over the results in [37].

5 Discussion

Next, we discuss a few further details and extensions.

Table 3: Classification accuracy for visual classification on the test set after the model parameter is tuned. Tuning achieves good improvements over the results in [37].

	Caltech101	Caltech256	Indoor67
[37]	87.22	67.23	56.79
ESTa	88.23	69.39	60.02
ESTn	88.25	69.39	60.08
	SUN397	Action40	Event8
[37]	42.61	54.92	94.42
ESTa	47.13	57.60	94.91
ESTn	47.21	57.58	94.86

Setting $\hat{m}$. In Section 2, we discussed one way of setting $\hat{m}$. There, we used an approximation with independent variables. We focused on Equations (3) and (4) throughout the paper since they yield upper bounds $\hat{m} \geq m$ that preserve our theoretical results. Nevertheless, other possibilities for setting $\hat{m}$ are conceivable. For example, close in spirit to Thompson and importance sampling, one may sample $\hat{m}$ from $\Pr[Y] = \prod_{\boldsymbol{x} \in \mathfrak{X}} \Phi(\frac{Y - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})})$. Furthermore, other search strategies can be used, such as guessing or doubling, and prior knowledge about properties of f such as the range can be taken into account.

Discretization. In our approximations, we used discretizations of the input space. While adding a bit more detail about this step here, we focus on the noiseless case, described by Equation (4). Equation (3) can be analyzed similarly for more general settings. For a Lipschitz continuous function, it is essentially sufficient to estimate the probability of $f(\boldsymbol{x}) \leq y, \forall \boldsymbol{x} \in \mathfrak{X}$ on set $\mathfrak{W}$, which is a ρ-covering of $\mathfrak{X}$.

We assume that f is Lipschitz continuous. Our analysis can be adapted to the milder assumption that f is Lipschitz continuous with high probability. Let L be the Lipschitz constant of f. By assumption, we have $|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leq L\rho$, for all $\|\boldsymbol{x} - \boldsymbol{x}'\| \leq \rho$. If $\mathfrak{X}$ is a continuous set, we construct
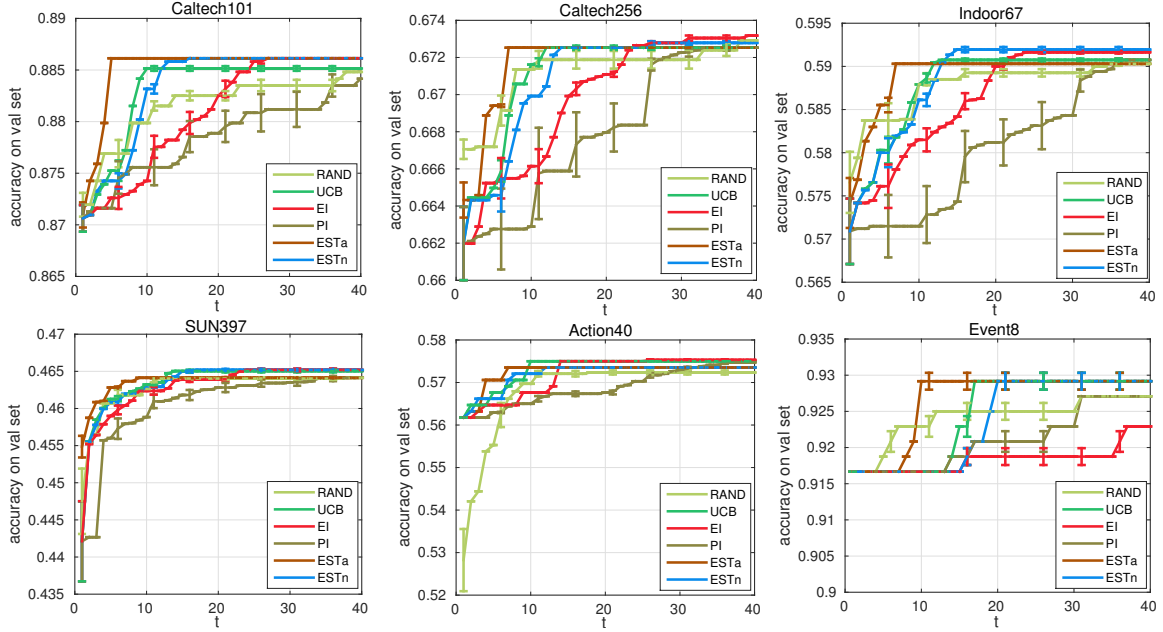
Figure 3: Maximum accuracy on the validation set over the iteration of the optimization. ESTa converges faster than other methods. Experiments are repeated 5 times, the standard deviation is scaled by 1/4.

its ρ-covering $\mathfrak{W}$ such that $\forall \boldsymbol{x} \in \mathfrak{X}, \inf_{\boldsymbol{x}' \in \mathfrak{W}} \|\boldsymbol{x} - \boldsymbol{x}'\| \leq \rho$. Let $E_{\mathfrak{X}}(y)$ be the event that $f(\boldsymbol{x}) \leq y, \forall \boldsymbol{x} \in \mathfrak{X}$. Then, $\Pr[E_{\mathfrak{X}}(y)] \geq \Pr[E_{\mathfrak{W}}(y - \rho L), E_{\mathfrak{X} \setminus \mathfrak{W}}(y)] = \Pr[E_{\mathfrak{W}}(y - \rho L)]$ The last step uses Lipschitz continuity to compute $\Pr[E_{\mathfrak{X} \setminus \mathfrak{W}}(y)|[E_{\mathfrak{W}}(y - \rho L)] = 1$. We can use this lower bound to compute $\hat{m}$, so $\hat{m}$ remains an upper bound on m. Notably, none of the regret bounds relies on a discretization of the space. Morever, once $\hat{m}$ is chosen, the acquisition function can be optimized with any search method, including gradient descent.

High dimensions. Bayesian Optimization methods generally suffer in high dimensions. Common assumptions are a low-dimensional or simpler underlying structure of f [7, 34, 15]. Our approach can be combined with those methods too, to be extended to higher dimensions.

Relation to entropy search. EST is closely related to entropy search (ES) methods [12, 13], but also differs in a few important aspects. Like EST, ES methods approximate the probability of a point $\boldsymbol{x}$ being the maximizer $\arg\max_{\boldsymbol{x}' \in \mathfrak{X}} f(\boldsymbol{x}')$, and then choose where to evaluate next by optimizing an acquisition function related to this probability. However, instead of choosing the input that is most likely to be the $\arg\max$, ES chooses where to evaluate next by optimizing the expected change in the entropy of $\Pr[M_{\boldsymbol{x}}]$. One reason is that ES does not aim to minimize cumulative regret like many other bandit methods (including EST). The cumulative regret penalizes all queried points, and a method that minimizes the cumulative regret

needs to query enough supposedly good points. ES methods, in contrast, purely focus on exploration, since their objective is to gather as much information as possible to estimate a final value in the very end. Since the focus of this work lies on cumulative regret, detailed empirical comparisons between EST and ES may be found in the supplement.

6 Conclusion

In this paper, we studied a new Bayesian optimization strategy derived from the viewpoint of the estimating the $\arg\max$ of an unknown function. We showed that this strategy corresponds to adaptively setting the trade-off parameters λ and θ in GP-UCB and GP-PI, and established bounds on the regret. Our experiments demonstrate that this strategy is not only easy to use, but robustly performs well by measure of different types of regret, on a variety of real-world tasks from robotics and computer vision.

References

[1] P. Auer. Using confidence bounds for exploitation-exploration tradeoffs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47 (2-3):235–256, 2002.

[3] E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. Technical Report TR-2009-023, University of British Columbia, 2009.

[4] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*, pages 23–37. Springer, 2009.

[5] A. D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12: 2879–2904, 2011.

[6] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth. An experimental comparison of Bayesian optimization for bipedal locomotion. In *International Conference on Robotics and Automation (ICRA)*, 2014.

[7] J. Djolonga, A. Krause, and V. Cevher. High-dimensional Gaussian process bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 2007.

[9] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on optimization*, 12(4):979–1006, 2002.

[10] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[11] S. Grünewälder, J.-Y. Audibert, M. Opper, and J. Shawe-Taylor. Regret bounds for Gaussian process bandit problems. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[12] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13:1809–1837, 2012.

[13] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[14] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[15] K. Kandasamy, J. Schneider, and B. Poczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning (ICML)*, 2015.

[16] A. Krause and C. S. Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[17] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Fluids Engineering*, 86(1):97–106, 1964.

[18] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *International Conference on Computer Vision (ICCV)*, 2007.

[19] D. J. Lizotte, T. Wang, M. H. Bowling, and D. Schuurmans. Automatic gait optimization with Gaussian process regression. In *International Conference on Artificial Intelligence (IJCAI)*, 2007.

[20] D. J. Lizotte, R. Greiner, and D. Schuurmans. An experimental methodology for response surface optimization methods. *Journal of Global Optimization*, 53(4):699–736, 2012.

[21] P. Massart. *Concentration Inequalities and Model Selection*, volume 6. Springer, 2007.

[22] J. Močkus. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, 1974.

[23] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[24] C. E. Rasmussen and C. K. Williams. Gaussian processes for machine learning. *The MIT Press*, 2006.

[25] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[26] A. M. Ross. Useful bounds on the expected maximum of correlated normal variables. Technical report, Technical Report 03W-004, ISE Dept., Lehigh Univ., Aug, 2003.

[27] J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow, and P. Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: Science and Systems Conference (RSS)*, volume 9, pages 1–10, 2013.

[28] D. Slepian. The one-sided barrier problem for Gaussian noise. *Bell System Technical Journal*, 41(2):463–501, 1962.

[29] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[30] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

[31] R. Tedrake. *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832)*. Downloaded in Fall, 2014 from http://people.csail.mit.edu/russt/underactuated/.

[32] R. Tedrake. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems. http://drake.mit.edu, 2014.

[33] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

[34] Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. De Freitas. Bayesian optimization in high dimensions via random embeddings. In *International Conference on Artificial Intelligence (IJCAI)*, 2013.

[35] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[36] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision (ICCV)*, 2011.

[37] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Supplement

In this supplement, we provide proofs for all theorems and lemmas in the main paper, more exhaustive experimental results and details on the experiments.

A Proofs

A.1 Proofs from Section 2

Lemma 2.1. *In any round t, the point selected by EST is the same as the point selected by a variant of GP-UCB with $\lambda_t = \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m}_t - \mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$. Conversely, the candidate selected by GP-UCB is the same as the candidate selected by a variant of EST with $\hat{m}_t = \max_{\boldsymbol{x} \in \mathfrak{X}} \mu_{t-1}(\boldsymbol{x}) + \lambda_t \sigma_{t-1}(\boldsymbol{x})$.*

Proof. We omit the subscripts t for simplicity. Let $\boldsymbol{a}$ be the point selected by GP-UCB, and $\boldsymbol{b}$ selected by EST. Without loss of generality, we assume $\boldsymbol{a}$ and $\boldsymbol{b}$ are unique. With $\lambda = \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$, GP-UCB chooses to evaluate

$$\boldsymbol{a} = \arg\max_{\boldsymbol{x} \in \mathfrak{X}} \mu(\boldsymbol{x}) + \lambda \sigma(\boldsymbol{x}) = \arg\min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}.$$

This is because

$$\hat{m} = \max_{\boldsymbol{x} \in \mathfrak{X}} \mu(\boldsymbol{x}) + \lambda \sigma(\boldsymbol{x}) = \mu(\boldsymbol{a}) + \lambda \sigma(\boldsymbol{a}).$$

By definition of $\boldsymbol{b}$, for all $\boldsymbol{x} \in \mathfrak{X}$, we have

$$\frac{\Pr[M_{\boldsymbol{b}}|\hat{m}, \mathfrak{D}]}{\Pr[M_{\boldsymbol{x}}|\hat{m}, \mathfrak{D}]} \approx \frac{Q(\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})}) \prod_{\boldsymbol{x}' \neq \boldsymbol{b}} \Phi(\frac{\hat{m}-\mu(\boldsymbol{x}')}{\sigma(\boldsymbol{x}')})}{Q(\frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}) \prod_{\boldsymbol{x}' \neq \boldsymbol{x}} \Phi(\frac{\hat{m}-\mu(\boldsymbol{x}')}{\sigma(\boldsymbol{x}')})}$$

$$= \frac{Q(\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})}) \Phi(\frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})})}{Q(\frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}) \Phi(\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})})}$$

$$\geq 1.$$

The inequality holds if and only if $\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})} \leq \frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$ for all $\boldsymbol{x} \in \mathfrak{X}$, including $\boldsymbol{a}$, and hence

$$\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})} \leq \frac{\hat{m}-\mu(\boldsymbol{a})}{\sigma(\boldsymbol{a})} = \lambda = \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})},$$

which, with uniqueness, implies that $\boldsymbol{a} = \boldsymbol{b}$ and GP-UCB and EST select the same point.

For the other direction, we denote the candidate selected by GP-UCB by

$$\boldsymbol{a} = \arg\max_{\boldsymbol{x} \in \mathfrak{X}} \mu(\boldsymbol{x}) + \lambda \sigma(\boldsymbol{x}).$$

The variant of EST with $\hat{m} = \max_{\boldsymbol{x} \in \mathfrak{X}} \mu(\boldsymbol{x}) + \lambda \sigma(\boldsymbol{x})$ selects

$$\boldsymbol{b} = \arg\max_{\boldsymbol{x} \in \mathfrak{X}} \Pr[M_{\boldsymbol{x}}|\hat{m}, \mathfrak{D}].$$

We know that for all $\boldsymbol{x} \in \mathfrak{X}$, we have $\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})} \leq \frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$ and hence $\hat{m} \leq \mu(\boldsymbol{b}) + \frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})} \sigma(\boldsymbol{b})$. Since $\hat{m} = \mu(\boldsymbol{a}) + \lambda \sigma(\boldsymbol{a})$, letting $\boldsymbol{x} = \boldsymbol{a}$ implies that

$$\hat{m} = \max_{\boldsymbol{x} \in \mathfrak{X}} \mu(\boldsymbol{x}) + \lambda \sigma(\boldsymbol{x}) \leq \mu(\boldsymbol{b}) + \lambda \sigma(\boldsymbol{b}).$$

Hence, by uniqueness it must be that $\boldsymbol{a} = \boldsymbol{b}$ and GP-UCB and EST select the same candidate. $\square$

A.2 Proofs from Section 3

Lemma 3.2. *Pick $\delta \in (0, 1)$ and set $\zeta_t = (2 \log(\frac{\pi_t}{2\delta}))^{\frac{1}{2}}$, where $\sum_{t=1}^{T} \pi_t^{-1} \leq 1$, $\pi_t > 0$. Then, for EST, it holds that $\Pr[\mu_{t-1}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t) \leq \zeta_t \sigma_{t-1}(\boldsymbol{x}_t)] \geq 1 - \delta$, for all $t \in [1, T]$.*

Proof. Let $z_t = \frac{\mu_{t-1}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t)}{\sigma_{t-1}(\boldsymbol{x}_t)} \sim \mathcal{N}(0, 1)$. It holds that

$$\Pr[z_t > \zeta_t] = \int_{\zeta_t}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \, \mathrm{d}z$$

$$= \int_{\zeta_t}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-(z-\zeta_t)^2/2 - \zeta_t^2/2 - z\zeta_t} \, \mathrm{d}z$$

$$\leq e^{-\zeta_t^2/2} \int_{\zeta_t}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-(z-\zeta_t)^2/2} \, \mathrm{d}z$$

$$= \frac{1}{2} e^{-\zeta_t^2/2}.$$

A union bound extends this bound to all rounds:

$$\Pr[z_t > \zeta_t \text{ for some } t \in [1, T]] \leq \sum_{t=1}^{T} \frac{1}{2} e^{-\zeta_t^2/2}.$$

With $\zeta_t = (2 \log(\frac{\pi_t}{2\delta}))^{\frac{1}{2}}$ and $\sum_{t=1}^{T} \pi_t^{-1} = 1$, this implies that with probability at least $1 - \delta$, it holds that $\mu_{t-1}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t) \leq \zeta_t \sigma_{t-1}(\boldsymbol{x}_t)$ for all $t \in [1, T]$. One may set $\pi_t = \frac{1}{6} \pi^2 t^2$, or $\pi_t = T$, in which case $\zeta_t = \zeta = (2 \log(\frac{T}{2\delta}))^{\frac{1}{2}}$. $\square$

Lemma 3.3. *If $\mu_{t-1}(\boldsymbol{x}_t) - f(\boldsymbol{x}_t) \leq \zeta_t \sigma_{t-1}(\boldsymbol{x}_t)$, the regret at time step t is upper bounded as $\tilde{r}_t \leq (\nu_t + \zeta_t) \sigma_{t-1}(\boldsymbol{x}_t)$, where $\nu_t \triangleq \min_{\boldsymbol{x} \in \mathfrak{X}} \frac{\hat{m}_t - \mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$, and $\hat{m}_t \geq \max_{\boldsymbol{x} \in \mathfrak{X}} f(\boldsymbol{x})$, $\forall t \in [1, T]$.*

Proof. At time step $t \geq 1$, we have

$$\tilde{r}_t = \max_{\boldsymbol{x} \in \mathfrak{X}} f(\boldsymbol{x}) - f(\boldsymbol{x}_t)$$

$$\leq \hat{m}_t - f(\boldsymbol{x}_t)$$

$$\leq \hat{m}_t - \mu_{t-1}(\boldsymbol{x}_t) + \zeta_t \sigma_{t-1}(\boldsymbol{x}_t)$$

$$= (\nu_t + \zeta_t) \sigma_{t-1}(\boldsymbol{x}_t).$$

$\square$

B Experiments

B.1 Approximate m

In the paper, we estimate m to be

$$\hat{m} = m_0 + \int_{m_0}^{\infty} 1 - \prod_{\boldsymbol{x} \in \mathfrak{W}} \Phi\left(\frac{w - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right) \mathrm{d}w$$

which involves an integration from the current maximum m_0 of the observed data to positive infinity. In fact the factor inside the integration quickly approaches zero in practice. We plot $g(w) = 1 - \prod_{\boldsymbol{x} \in \mathfrak{W}} \Phi\left(\frac{w - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right)$ in Figure 4, which looks like half of a Gaussian distribution. So instead of numerical integration (which can be done efficiently), heuristically we can sample two values of $g(w)$ to fit $\hat{g}(w) = ae^{-(w-m_0)^2/2b^2}$ and do the integration $\int_{m_0}^{\infty} \hat{g}(w) \, \mathrm{d}w = \sqrt{2\pi}ab$ analytically to be more efficient. This method is what we called ESTa in the paper, while the numerical integration is called ESTn.

We notice that our estimation $\hat{m}$ can serve as a tight upper bound on the real value of the max of the function in practice. One example of is shown in Figure 4 with a 1-D GP function. This example shows how PI, ESTa and ESTn estimate m. Both ESTa and ESTn are upper bounds of the true maximum of the function, and ESTn is actually very tight. For PI, $\theta = \arg\max_{1 \leq \tau < t} y_\tau + \epsilon$ is always a lower bound of an ϵ shift over the true maximum of the function.

Figure 4: Top: $g(w)$, $w \in [m_0, +\infty)$; Bottom: estimation of m.

B.2 Synthetic data

We show the examples of the functions we sampled from GP in Figure 5. The covariance function of GP is an isotropic Matérn kernel with parameters $\ell = 0.1, \sigma_f = 1$. The mean function is a linear function with a fixed random slope for different dimensions, and the constant is 1.

B.3 Initialization tuning for trajectory optimization

The 8 configurations of start state are $[7\ 1\ 0\ 0]$, $[7\ 0\ 0\ 0]$, $[1\ 0\ 0\ 0]$, $[1\ 1\ 0\ 0]$, $[2\ 0\ 0\ 0]$, $[2\ 1\ 0\ 0]$, $[3\ 0\ 0\ 0]$, $[3\ 1\ 0\ 0]$, where the first two dimensions denote the position and the last two dimension denote the speed. We only tune the first two dimension and keep the speed to be 0 for both directions. The target state is fixed to be $[5\ 9\ 0\ 0]$.

We can initialize the trajectory by setting the mid point of trajectory to be any point falling on the grid of the space (both x axis and y axis have range $[-2, 12]$). Then use SNOPT to solve the trajectory optimization problem, which involves an objective cost function (we take the negative cost to be a reward function to maximize), dynamics constraints, and obstacle constraints etc. Details of trajectory optimization are available in [32, 31].
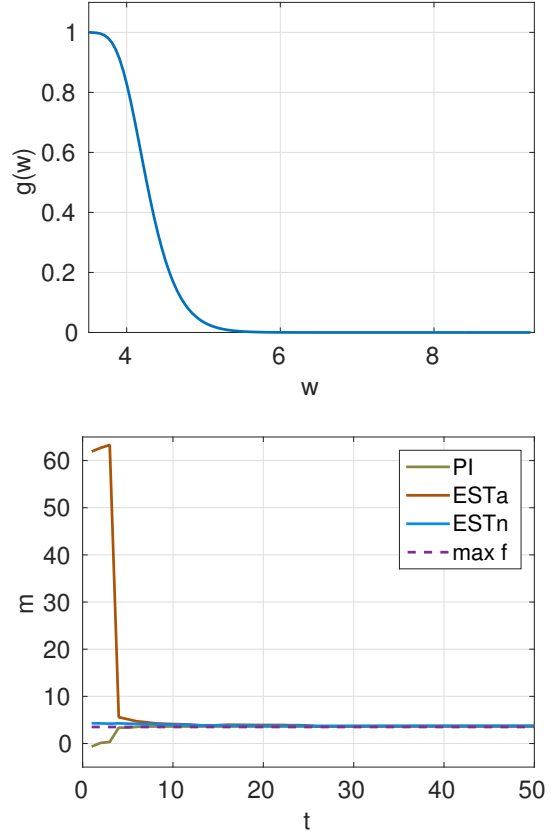
We used the same settings of parameters for GP as in Section B.2 for all the methods we tested and did kernel parameter fitting every 5 rounds. The same strategy was used for the image classification experiments in the next section.

B.4 Parameter tuning for image classification

We use the linear SVM in the liblinear package for all the image classification experiments. We extract the FC7 activation from the imagenet reference network in the Caffe deep learning package [14] as the visual feature. The reported classification accuracy is the accuracy averaged over all the categories. '-c' cost is the model parameter we tune for the linear SVM.

In Caltech101 and Caltech256 experiment [8, 10], there are 8,677 images from 101 object categories in the Caltech101 and 29,780 images from 256 object categories. The training size is 30 images per category, and the rest are test images.

In SUN397 experiment [35], there are 108,754 images from 397 scene categories. Images are randomly split into training and test set. The training size is 50 images per category, and the rest are test images.
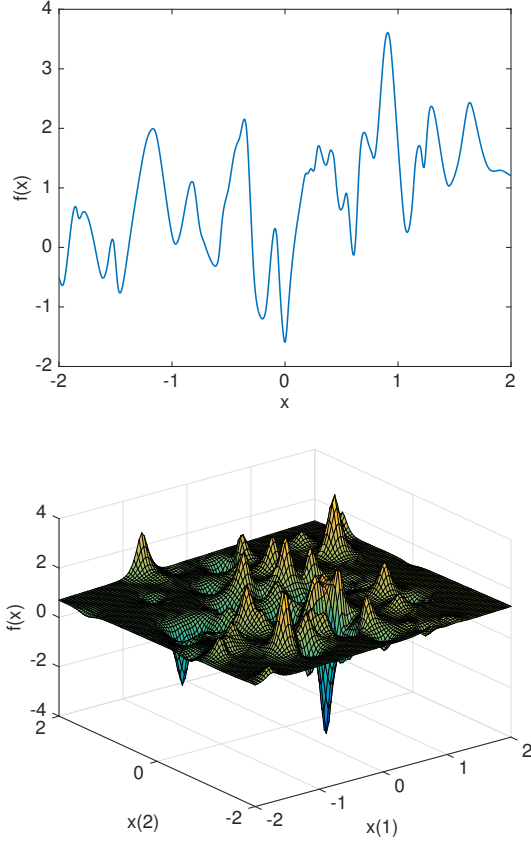
Figure 5: Examples of a function sampled from 1-D GP (top), and a function sampled from 2-D GP (bottom) with isotropic Matérn kernel and linear mean function. We deliberately create many local optimums to make the problem hard.

In MIT Indoor67 experiment [23], there are 15,620 images from 67 indoor scene categories. Images are randomly split into training set and test set. The training size is 100 images per category, and the rest are test images.

In Stanford Action40 experiment [36], there are 9,532 images from 40 action categories. Images are randomly split into training set and test set. The training size is 100 images per category, and the rest are test images.

In UIUC Event8 experiment [18], there are 1,579 images from 8 event categories. Images are randomly split into training set and test set. Training size is 70 images per category, and the rest are test images.

We used features extracted from a convolutional neural network (CNN) that was trained on images from ImageNet. It has been found [37] that features from a CNN trained on a set of images focused more on places than on objects, the Places database, work better in some domains. So, we repeated our experiments using the Places-CNN features, and the results are shown in Figure 6 and Table 4.

All the methods help to improve the classification accuracy on the validation set. EST methods achieve good accuracy on each validation set on par with the best competitors for most of the datasets. And we also observe that for Caltech101 and Event8, RAND and UCB converge faster and achieve better accuracy than other methods. As we have shown in Section 4.1, UCB and RAND perform worse than other methods in terms of cumulative regret, because they tend to explore too much. However, more exploration can be helpful for some black-box functions that do not satisfy our assumption that they are samples from GP. For example, for discontinuous step functions, pure exploration can be beneficial for simple regret. One possible explanation for the better results of RAND and UCB is that the black-box functions we optimize here are possibly functions not satisfying our assumption. The strong assumption on the black-box function is also a major drawback of Bayesian optimization,

B.5 Comparison to entropy search methods

Entropy search methods [12, 13] aim to minimize the entropy of the probability for the event $M_{\boldsymbol{x}}$ ($\boldsymbol{x} = \arg\max_{\boldsymbol{x}' \in \mathfrak{X}} f(\boldsymbol{x}')$). Although not suitable for minimizing cumulative regret, ES methods are intuitively ideal for minimizing simple regret. We hence in this section compare the empirical performance of entropy search (ES) [12] and predictive entropy search (PES) [13] to that of the EST methods (EST/GP-UCB/PI) and EI.

Since both ES and PES only support squared exponential covariance function and zero mean function in their code right now, and it requires significant changes in their code to accommodate other covariance functions, we created synthetic functions that are different from the ones we used in Section 4 in the paper. The new functions are sampled from 1-D (80 functions) and 2-D GP (20 functions) with squared exponential kernel ($\sigma_f = 0.1$ and $l = 1$) and 0 mean. Function examples are shown in Figure 9.

We show the results on these synthetic functions in Figure 7,8, and a standard optimization test function, Branin Hoo function, in Figure 10. It is worth noting that ES methods make queries on the most informative points, which are not necessarily the points with low regret. At each round, ES methods make a "query" on the black-box function, and then make a "guess" of the $\arg\max$ of the function (but do not test the "guess"). We plot the regret achieved by the "guesses" made by ES methods. For the 1-D GP task, all the methods behave similarly and achieve zero regret except RAND. For the 2-D GP task, EI is the fastest method to converge to zero regret, and in the end ESTn, PI,EI and ES methods achieve similar results. For the test on Branin Hoo function, PES achieves the lowest regret. ESTa converges slightly faster than PES, but to a slightly higher regret.

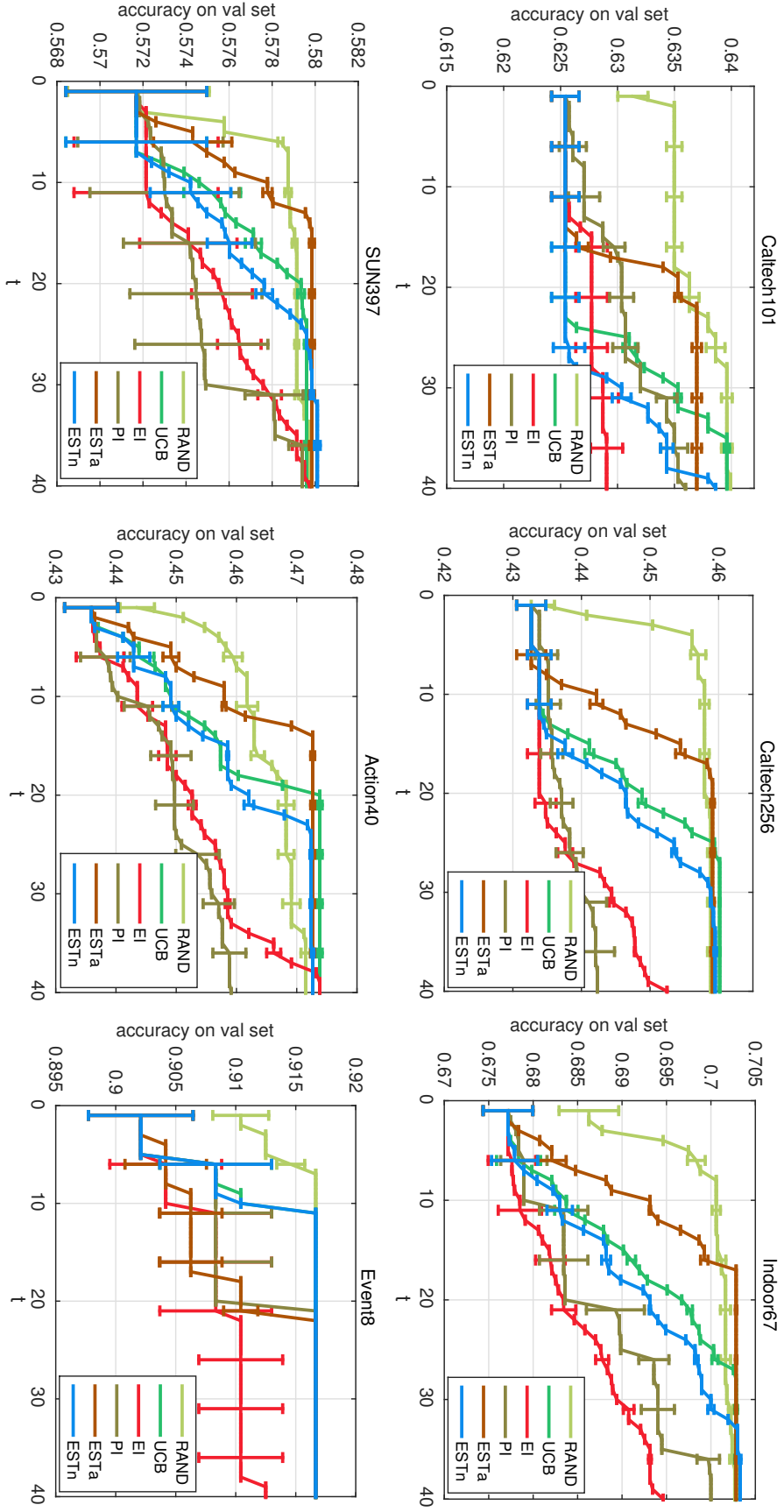We also compared the running time for all the methods in

Figure 6: Maximum accuracy on the validation set over iteration of the optimization. Experiments are repeated 5 times. The visual features used here are Deep CNN features pre-trained on the Places database.

Table 4: Classification accuracy on the test set of the datasets after the model parameter is tuned by ESTa and ESTn. Tuning achieves good improvement over the results in [37].

	Caltech101	Caltech256	Indoor67	SUN397	Action40	Event8
Imagenet-CNN feature	87.22	67.23	56.79	42.61	54.92	94.42
ESTa	88.23	69.39	60.02	47.13	57.60	94.91
ESTn	88.25	69.39	60.08	47.21	57.58	94.86
Places-CNN feature	65.18	45.59	68.24	54.32	42.86	94.12
ESTa	66.94	47.51	70.27	58.57	46.24	93.79
ESTn	66.95	47.43	70.27	58.65	46.17	93.56
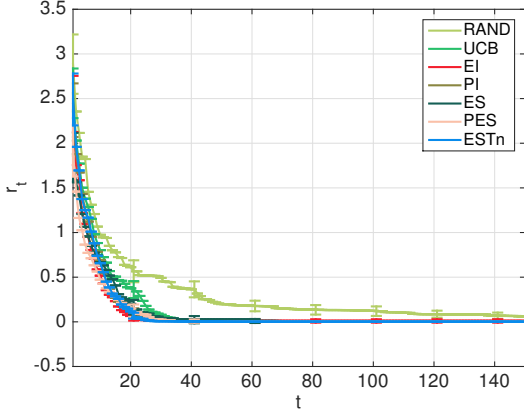
Figure 7: Simple regret for functions sampled from 1-D GP with squared exponential kernel and 0 mean.
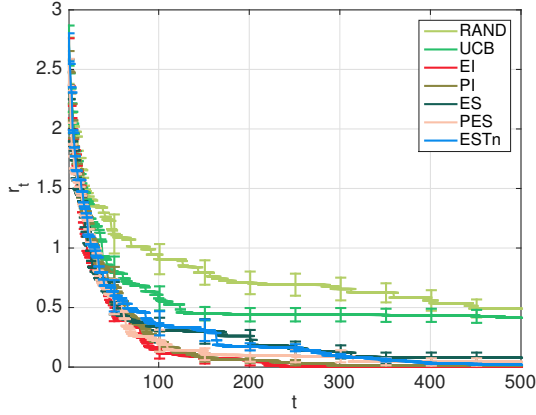

Figure 8: Simple regret for functions sampled from 2-D GP with squared exponential kernel and 0 mean.
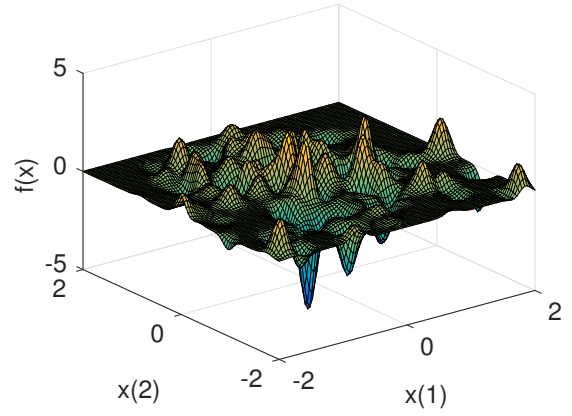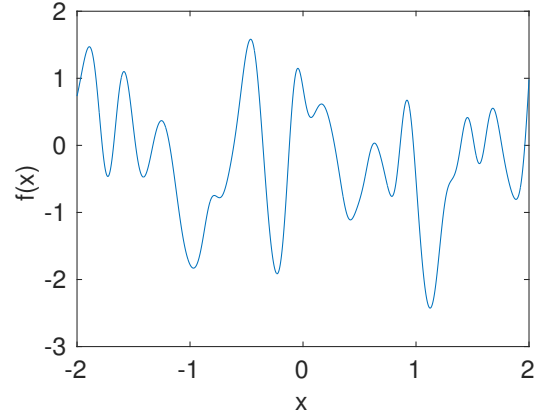

Figure 9: Examples of a function sampled from 1-D GP (left), and a function sampled from 2-D GP (right) with squared exponential kernel and 0 mean functions. These functions can be easier than the ones in Figure 5 since they have fewer local optima.

Table 5. [3] It is assumed in GP optimization that it is more expensive to evaluate the blackbox function than computing the next query to evaluate using GP optimization techniques. However, in practice, we still want the algorithm to output the next query point as soon as possible. For ES methods, it can be sometimes unacceptable to run them for black-box functions that take minutes to complete a query.

_____

[3]All of the methods were run with MATLAB (R2012b), on Intel(R) Xeon(R) CPU E5645 @ 2.40GHz.

Table 5: Comparison on the running time (s) per iteration.

RAND	UCB	EI	PI
0.0002	0.075	0.079	0.076

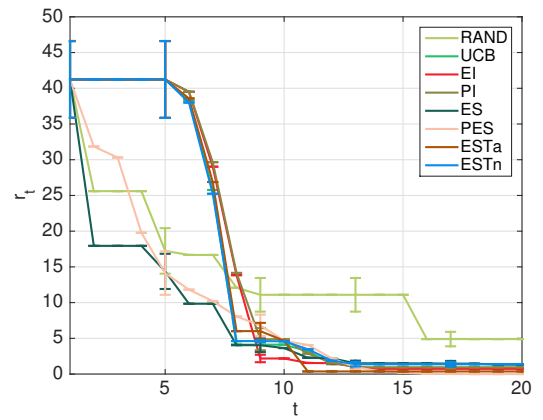| ESTa	ESTn	ES	PES
0.078	0.55	56	106

Figure 10: Simple regret for Branin Hoo function. UCB/EI/PI/ESTa/ESTn use the isotropic Matérn kernel with parameters $\ell = 0.1, \sigma_f = 1$; ES/PES use the isotropic squared exponential kernel with $\ell = 0.1, \sigma_f = 1$.