Optimization as Estimation with Gaussian Processes in Bandit Settings

Anonymous Authors
Affiliation

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 a maximum a posteriori (MAP) 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. The MAP criterion 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. An increasingly popular direction has been to model smoothness assumptions on the function via a Gaussian Process (GP). This probabilistic approach provides a posterior distribution of the unknown function, and thereby uncertainty estimates that help to decide where to evaluate the function next, in search of a maximum. Recent successful applications of this *Bayesian optimization* framework include hyperparameter tunning for algorithms in machine learning, robotics, and computer vision [3, 6, 16, 19, 29, 33].

Despite progress on theory and applicability of Bayesian optimization methods, the practitioner continues to face open ends: there is a menu of algorithms, and their relations and tradeoffs are only partially understood. Popular algorithms include *probability of improvement* (PI) [17], *expected improvement* (EI) [22], *upper confidence bound* (UCB) [30], and *entropy search* (ES) [12]. 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].

The motivations and analyses (if available) differ too: objectives include *cumulative regret*, *simple regret*, the performance under a fixed finite budget [11], or the uncertainty about the location of the maximizing argument [12]. Some algorithms perform well for some goals but not for others. Ideally, we would like an algorithm that in practice performs robustly for a range of these goals without time-consuming tuning of its own parameters.

Motivated by these practical concerns, in particular for applications with expensive function evaluations in robotics and computer vision, we propose a practical optimization strategy that does not require setting any parameters, saving the tuning time of the Bayesian optimization strategy. Our strategy relies on MAP inference of the argmax of the unknown function. We show connections between MAP and the popular GP-UCB and GP-PI strategies, implying a clear, intuitive, and provably correct way of setting the parameters adaptively in those important methods. Moreover, we establish bounds on the regret of the MAP estimation strategy. Our thorough empirical evaluation

includes problems from non-convex optimization, robotics, and computer vision. It suggests that the MAP strategy is not only practically easy to use, but is also robustly competitive for a number of performance criteria and applications: different types of regret, as well as the number of steps to reach a fixed regret value.

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 $\Phi(\cdot)$ the cumulative probability of $\phi(\cdot)$.

In round t of a bandit game, the player chooses $\boldsymbol{x}_t$ and then observes $y_t = f(\boldsymbol{x}_t) + \epsilon_t$. We define the *regret* for round t 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 $\boldsymbol{x}_t = \arg\max_{\boldsymbol{x} \in \mathfrak{X}} \mu_{t-1}(\boldsymbol{x}) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(\boldsymbol{x})$ where $\beta_t = 2 \log(|\mathfrak{X}| \pi^2 t^2 / 6\delta)$ for finite $\mathfrak{X}$. Their regret bound holds with probability $1 - \delta$. In practice, δ must be selected.

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

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

2 MAP estimation for optimization

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 via MAP inference. We will see how, as a side effect, this criterion establishes connections between the above criteria. The MAP strategy eventually leads to tighter bounds than GP-UCB as shown in Section C.

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 $\Pr[M_{\boldsymbol{x}} | \mathfrak{D}] = \Pr[\forall \boldsymbol{x}' \in \mathfrak{X}, v(\boldsymbol{x}') \leq 0 | \mathfrak{D}]$. 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 appendix). So $\Pr[M_{\boldsymbol{x}} | \mathfrak{D}]$ 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 $\{f(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathfrak{W}}$ by independent Gaussian random variables with means $\mu(\boldsymbol{x})$ and variances $\sigma(\boldsymbol{x})^2$ for all $\boldsymbol{x}\in\mathfrak{W}$ and $\mathfrak{W}=\mathfrak{X}\setminus\mathfrak{D}$.

We either assume knowledge of the true maximum m of the function, or estimate it via the posterior expectation of $Y=\max_{\boldsymbol{x}\in\mathfrak{X}} f(\boldsymbol{x})$ given $\mathfrak{D}$ [26]:

$$\hat{m}=\mathbb{E}[Y|\mathfrak{D}]=\int_0^\infty \Pr[Y>y|\mathfrak{D}]-\Pr[Y<-y|\mathfrak{D}]\,\mathrm{d}y. \tag{1}$$

In the noiseless case, this becomes $\hat{m}=m_0+\int_{m_0}^\infty 1-\prod_{\boldsymbol{x}\in\mathfrak{W}}\Phi\Big(\frac{w-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\Big)\,\mathrm{d}w$ where $m_0=\max_{\tau\in[1,t-1]} y_\tau$ is the current observed maximum value. Under conditions specified in Section C, the mean field approximation 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 D.

Knowing m, the event $M_{\boldsymbol{x}}|m,\mathfrak{D}$ is equivalent to

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

The MAP estimation strategy (EST) plays $\boldsymbol{x}_t=\arg\max_{\boldsymbol{x}\in\mathfrak{X}}\Pr[M_{\boldsymbol{x}}|\hat{m},\mathfrak{D}_t]$. We discuss the continuous settings in Section D.

Next, we relate our strategy to GP-PI and GP-UCB: EST turns out to be equivalent to adaptively tuning θ in GP-PI and $\beta_t^{\frac{1}{2}}$ in GP-UCB. This observation reveals unifying connections between GP-PI and GP-UCB and, in Section C, yields regret bounds for GP-PI with a certain choice of θ. Lemma 2.1 characterizes the connection to GP-UCB:

Lemma 2.1. *At any step t, the point selected by EST is the same as the point selected by a variant of GP-UCB with $\beta_t^{\frac{1}{2}}=\min_{\boldsymbol{x}\in\mathfrak{X}}\frac{\hat{m}-\mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$. Conversely, the candidate selected by UCB is the same as the candidate selected by a variant of EST with $\hat{m}=\max_{\boldsymbol{x}\in\mathfrak{X}}\mu_{t-1}(\boldsymbol{x})+\beta_t^{\frac{1}{2}}\sigma_{t-1}(\boldsymbol{x})$.*

Proposition 2.2. *GP-PI is equivalent to EST when setting $\theta=\hat{m}$ 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 $\beta^{\frac{1}{2}}$ is set to $\min_{\boldsymbol{x}\in\mathfrak{X}}\frac{\theta-\mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$, and GP-PI corresponds to GP-UCB if $\theta=\max_{\boldsymbol{x}\in\mathfrak{X}}\mu_{t-1}(\boldsymbol{x})+\beta_t^{\frac{1}{2}}\sigma_{t-1}(\boldsymbol{x})$.*

Note that there still remain important differences between GP-PI and EST, as shown in the following theorem which establishes EST's regret bounds. More details are available in Section C.

Theorem 2.4. *Let σ^2 be the variance of the Gaussian noise in the observation, $C=2/\log(1+\sigma^{-2})$, γ_T the maximum information gain of the selected points, and $t^*=\arg\max_t\nu_t$ where $\nu_t^{\frac{1}{2}}\triangleq\min_{\boldsymbol{x}\in\mathfrak{X}}\frac{m-\mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$. With probability $1-\delta$, the cumulative regret up to time step T is bounded as $R_T=\sum_{t=1}^T\tilde{r}_t\leq\sqrt{CT\gamma_T}(\nu_{t^*}^{\frac{1}{2}}+\zeta_T^{\frac{1}{2}})$, where ζ_T is defined in Lemma C.1.*

3 Experiments

In this section, we show experimental results on synthetic black box functions and parameter tuning for image classification. We compare the following methods: EST with a Laplace approximation, i.e., approximating the integrand in Eqn. (1) by a truncated Gaussian (ESTa, details in supplement); EST with numerical integration to evaluate Eqn. (1) (ESTn); UCB; EI; PI; and random selection (RAND). We omit the 'GP-' prefix for simplicity. Additional experimental details and results on initialization tuning for trajectory optimization may be found in Section E.

3.1 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.

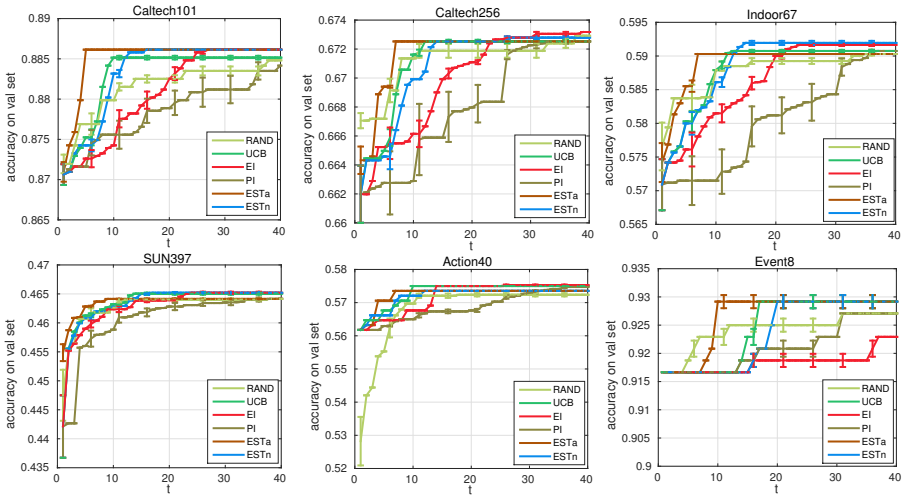

Figure 1: Maximum accuracy on the validation set over the iteration of the optimization. ESTa converges faster than other methods. Experiments are repeated 5 times, the standard deviation is scaled by 1/4.

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 $\hat{r}_{\min}$). RAND is inferior both in terms of $T_{\min}$ and $r_{\min}$. Although EI and PI behaves very well in terms of T_{min}, they cannot achieve the regret as low as UCB and the two methods of EST. UCB finds a good point but takes more than twice as many rounds as EST for doing so. The results for synthetic functions suggest that throughout, compared to other methods, EST finds better function values within a smaller number of iterations.

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

Table 1: Experimental results on functions sampled from 1-D GP (top) and 2-D GP (bottom) with a limited budget of iterations. '–' denotes the mean and '^' the median.

3.2 Parameter Tuning for Image Classification

The experiments in this section addresses Bayesian optimization for efficiently tuning parameters in visual classification tasks. Here, $\boldsymbol{x}_t$ is the candidate 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]. Figure 1 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.

4 Conclusion

In this paper, we studied a new Bayesian optimization strategy derived from the viewpoint of MAP estimation. We showed that this strategy corresponds to adaptively setting the trade-off parameters β 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.

References

[1] P. Auer. Using confidence bounds for exploitation-exploration tradeoffs. *JMLR*, 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. 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. Bull. Convergence rates of efficient global optimization algorithms. *JMLR*, 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 *ICRA*, 2014.

[7] J. Djolonga, A. Krause, and V. Cevher. High-dimensional Gaussian process bandits. In *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. 2007.

[11] S. Grünewälder, J.-Y. Audibert, M. Opper, and J. Shawe-Taylor. Regret bounds for Gaussian process bandit problems. In *AISTATS*, 2010.

[12] P. Hennig and C. Schuler. Entropy search for information-efficient global optimization. *JMLR*, 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 *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 *ICML*, 2015.

[16] A. Krause and C. S. Ong. Contextual Gaussian process bandit optimization. In *NIPS*, 2011.

[17] H. Kushner. A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise. *J. Basic Eng.*, 86:97–106, 1964.

[18] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *ICCV*, 2007.

[19] D. Lizotte, T. Wang, M. Bowling, and D. Schuurmans. Automatic gait optimization with Gaussian process regression. In *IJCAI*, 2007.

[20] D. 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 *IFIP technical conference*, 1974.

[23] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009.

[24] C. E. Rasmussen. Gaussian processes for machine learning. 2006.

[25] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *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 *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 *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 *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, 2014. URL `http://drake.mit.edu`.

[33] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In *KDD*, 2013.

[34] Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. de Freitas. Bayesian optimization in high dimensions via random embeddings. In *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 *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 *ICCV*, 2011.

[37] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.

A 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 its variant predictive entropy search (PES) [13] do not aim to minimize regret directly, but to maximize the amount of information gained about the optimal point. Their queries made at each iteration are different from their guesses for the optimal point. Thus entropy search methods are not suitable for bandit settings. 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]; the finite-budget strategy in [11] is mainly of theoretical interest. In this work, in contrast, we are motivated by practical considerations.

B Implementation

Algorithm 1 GP-UCB/PI/EST

$\mathfrak{D} \leftarrow \emptyset$
while stopping criterion not reached **do**
  $\mu, \Sigma \leftarrow$ GP-predict($\mathfrak{X}|\mathfrak{D}$)
  $\hat{m} = \begin{cases} \max_{\boldsymbol{x}\in\mathfrak{X}} \mu(\boldsymbol{x}) + \beta_t^{\frac{1}{2}} \sigma(\boldsymbol{x}) & \text{GP-UCB} \\ \max_{1\le\tau<t} y_\tau + \epsilon & \text{GP-PI} \\ \mathbb{E}[Y|\mathfrak{D}] \text{ (see Eqn. (1))} & \text{GP-EST} \end{cases}$
  $\boldsymbol{x}_t \leftarrow \arg\min_{\boldsymbol{x}\in\mathfrak{X}} \frac{\hat{m}-\mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$
  $y_t \leftarrow f(\boldsymbol{x}_t)$
  $\mathfrak{D} \leftarrow \{\boldsymbol{x}_\tau, y_\tau\}_{\tau=1}^t$
  $t \leftarrow t+1$
end while
$\boldsymbol{x}^* \leftarrow \arg\max_{(\boldsymbol{x},y)\in\mathfrak{D}} y_t$

Proposition 2.2 suggests that we need not calculate the probability $\Pr[M_{\boldsymbol{x}}|\hat{m}, \mathfrak{D}]$ directly when implementing GP-EST. Instead, we can reduce it to GP-PI with an automatically tuned target value

θ. Algorithm 1 compares the pseudocode for all three methods. GP-UCB/PI/EST all share the same idea of reaching a target value ($\hat{m}$ 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 $\max_{\boldsymbol{x}\in\mathfrak{X}} \mu(\boldsymbol{x})+\beta_t^{\frac{1}{2}}\sigma(\boldsymbol{x})$ with $\beta_t = 2\log(|\mathfrak{X}|\pi^2 t^2/6\delta)$, as a result of applying the union bound on all the members of $\mathfrak{X}$[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. GP-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 GP-EST is not only intuitively reasonable, but it also brings about great improvement over GP-UCB/PI in terms of the regret bounds.

C Proofs

In this section, we provide the proof for Lemma 2.1.

Lemma 2.1. *At any step t, the point selected by EST is the same as the point selected by a variant of GP-UCB with* $\beta_t^{\frac{1}{2}} = \min_{\boldsymbol{x}\in\mathfrak{X}} \frac{\hat{m}-\mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$. *Conversely, the candidate selected by UCB is the same as the candidate selected by a variant of EST with* $\hat{m} = \max_{\boldsymbol{x}\in\mathfrak{X}} \mu_{t-1}(\boldsymbol{x}) + \beta_t^{\frac{1}{2}}\sigma_{t-1}(\boldsymbol{x})$.

Proof. We omit the subscripts t for simplicity. Let $\boldsymbol{a}$ be the point selected by GP-UCB, and $\boldsymbol{b}$ the one selected by EST. Without loss of generality, we assume $\boldsymbol{a}$ and $\boldsymbol{b}$ are unique. With $\beta^{\frac{1}{2}} = \min_{\boldsymbol{x}\in\mathfrak{X}} \frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$, GP-UCB plays $\boldsymbol{a} = \max_{\boldsymbol{x}\in\mathfrak{X}} \mu(\boldsymbol{x}) + \beta_t^{\frac{1}{2}}\sigma(\boldsymbol{x}) = \arg\min_{\boldsymbol{x}\in\mathfrak{X}} \frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$, because

$$\hat{m} = \max_{\boldsymbol{x}\in\mathfrak{X}} \mu(\boldsymbol{x}) + \beta^{\frac{1}{2}}\sigma(\boldsymbol{x}) = \mu(\boldsymbol{a}) + \beta^{\frac{1}{2}}\sigma(\boldsymbol{a})$$

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

$$1 \le \frac{\Pr[M_{\boldsymbol{b}}|\hat{m},\mathfrak{D}]}{\Pr[M_{\boldsymbol{x}}|\hat{m},\mathfrak{D}]} = \frac{Q(\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})})\prod_{\boldsymbol{x}'\neq\boldsymbol{b}}\Phi(\frac{\hat{m}-\mu(\boldsymbol{x}')}{\sigma(\boldsymbol{x}')})}{Q(\frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})})\prod_{\boldsymbol{x}'\neq\boldsymbol{x}}\Phi(\frac{\hat{m}-\mu(\boldsymbol{x}')}{\sigma(\boldsymbol{x}')})} = \frac{Q(\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})})\Phi(\frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})})}{Q(\frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})})\Phi(\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})})} \tag{2}$$

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

$$\frac{\hat{m}-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})} \le \frac{\hat{m}-\mu(\boldsymbol{a})}{\sigma(\boldsymbol{a})} = \beta^{\frac{1}{2}} = \min_{\boldsymbol{x}\in\mathfrak{X}} \frac{\hat{m}-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})} \tag{3}$$

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

For the other direction, denote the candidate selected by UCB by $\boldsymbol{a} = \arg\max_{\boldsymbol{x}\in\mathfrak{X}} \mu(\boldsymbol{x}) + \beta^{\frac{1}{2}}\sigma(\boldsymbol{x})$. The variant of EST with $m = \max_{\boldsymbol{x}\in\mathfrak{X}} \mu(\boldsymbol{x}) + \beta^{\frac{1}{2}}\sigma(\boldsymbol{x})$ selects $\boldsymbol{b} = \arg\max_{\boldsymbol{x}\in\mathfrak{X}} \Pr[M_{\boldsymbol{x}}|m,\mathfrak{D}]$. We know that for all $\boldsymbol{x}\in\mathfrak{X}$, we have $\frac{m-\mu(\boldsymbol{b})}{\sigma(\boldsymbol{b})} \le \frac{m-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$ and hence $m \le \mu(\boldsymbol{b}) + \frac{m-\mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\sigma(\boldsymbol{b})$. Since $m = \mu(\boldsymbol{a}) + \beta^{\frac{1}{2}}\sigma(\boldsymbol{a})$, letting $\boldsymbol{x} = \boldsymbol{a}$ implies that

$$m = \max_{\boldsymbol{x}\in\mathfrak{X}} \mu(\boldsymbol{x}) + \beta^{\frac{1}{2}}\sigma(\boldsymbol{x}) \le \mu(\boldsymbol{b}) + \beta^{\frac{1}{2}}\sigma(\boldsymbol{b}). \tag{4}$$

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

C Regret Bounds

In this section, we analyze the regret of EST. We first show a high-probability bound, and then a bound on the expected regret. For the moment, we assume knowledge of m, but the analysis in the following still holds by replacing m with $\hat{m}$ if $\hat{m} \ge m$. We show that this condition is satisfied if we use $\hat{m}$ in Eqn. (1) and Eqn. (2) with mean field assumption under the condition specified in Corollary C.6.

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

Theorem 2.4. *Let σ^2 be the variance of the Gaussian noise in the observation, $C = 2/\log(1+\sigma^{-2})$, γ_T the maximum information gain of the selected points, and $t^* = \arg\max_t \nu_t$ where $\nu_t^{\frac{1}{2}} \triangleq \min_{\boldsymbol{x}\in\mathfrak{X}} \frac{m-\mu_{t-1}(\boldsymbol{x})}{\sigma_{t-1}(\boldsymbol{x})}$. With probability $1-\delta$, the cumulative regret up to time step T is bounded as $R_T = \sum_{t=1}^T \tilde{r}_t \leq \sqrt{CT\gamma_T}(\nu_{t^*}^{\frac{1}{2}} + \zeta_T^{\frac{1}{2}})$, where ζ_T is defined in Lemma C.1.*

The information gain γ_T after T rounds is the maximum mutual information that can be gained about f from T measurements: $\gamma_T = \max_{A\subseteq\mathfrak{X},|A|\leq T} I(\boldsymbol{y}_A, \boldsymbol{f}_A) = \max_{A\subseteq\mathfrak{X},|A|\leq T} \frac{1}{2}\log\det(\boldsymbol{I} + \sigma^{-2}\boldsymbol{K}_A)$. Without loss of generality, assume $k(x,x)\leq 1$ and $\mathfrak{X}\subset\mathbb{R}^d$; then 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 ξ is the roughness parameter of the kernel [30, Theorem 5].

The proof of Theorem 2.4 follows [30] and relies on the following lemmas which are proved in the supplement.

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

Proof. The proof is similar to Lemma 5.1 in [30, Appendix A.1]. We can set $\pi_t = \frac{\pi^2 t^2}{6}$ (e.g. if T is unknown and possibly infinite) or $\pi_t = T$, in which case $\zeta_t = \zeta = 2\log(T/\delta)$ is fixed. Here we emphasize that $\boldsymbol{x}_t$ is determined by the strategy we use to pick it, so $\boldsymbol{x}_t$ is **not** a random vector. Thus it does not need to apply union bound on every member of $\mathfrak{X}$ since the union bound on $\{\boldsymbol{x}_t\}_{t=1}^T$ is sufficient. □

Lemma C.2. *If $|f(\boldsymbol{x}_t) - \mu_{t-1}(\boldsymbol{x}_t)| \leq \zeta_t^{\frac{1}{2}}\sigma_{t-1}(\boldsymbol{x}_t)$, the regret at time step t is upper bounded by $\tilde{r}_t \leq (\nu_t^{\frac{1}{2}} + \zeta_t^{\frac{1}{2}})\sigma_{t-1}(\boldsymbol{x}_t)$, and hence $\min_{t\in[1,T]}\tilde{r}_t \leq (\nu_T^{\frac{1}{2}} + \zeta_T^{\frac{1}{2}})\sigma_{T-1}(\boldsymbol{x}_T)$.*

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

$$\tilde{r}_t \leq m - f(\boldsymbol{x}_t) \tag{5}$$

$$\leq m - \mu_{t-1}(\boldsymbol{x}_t) + \zeta_t^{\frac{1}{2}}\sigma_{t-1}(\boldsymbol{x}_t) \tag{6}$$

$$= (\nu_t^{\frac{1}{2}} + \zeta_t^{\frac{1}{2}})\sigma_{t-1}(\boldsymbol{x}_t) \tag{7}$$

For the minimum regret, we can bound

$$\min_t \tilde{r}_t \leq \tilde{r}_T = (\nu_T^{\frac{1}{2}} + \zeta_T^{\frac{1}{2}})\sigma_{T-1}(\boldsymbol{x}_T)$$ □

The proof of Theorem 2.4 now follows from Lemmas C.1 and C.2.

Proof. (Thm. 2.4) By Lemma C.2, we have that $\tilde{r}_t^2 \leq (\nu_{t^*}^{\frac{1}{2}} + \zeta_T^{\frac{1}{2}})^2\sigma_{t-1}^2(\boldsymbol{x}_t)$. Using $(1+a)^x \leq 1+ax$ if $0\leq x\leq 1$, it follows that

$$\sigma_{t-1}^2(\boldsymbol{x}_t) \leq \frac{\log(1+\sigma^{-2}\sigma_{t-1}^2(\boldsymbol{x}_t))}{\log(1+\sigma^{-2})}.$$

By Lemma 5.3 and Lemma 5.4 in [30], we further bound $\sum_{t=1}^T \sigma_{t-1}^2(\boldsymbol{x}_t) \leq \frac{2}{\log(1+\sigma^{-2})}\gamma_T$. Now, by the Cauchy-Schwarz inequality, we obtain

$$\sum_{t=1}^T \sigma_{t-1}(\boldsymbol{x}_t) \leq \sqrt{T\sum_{t=1}^T \sigma_{t-1}^2(\boldsymbol{x}_t)} \leq \sqrt{\frac{2T}{\log(1+\sigma^{-2})}\gamma_T}.$$

and therefore

$$\sum_{t=1}^T \tilde{r}_t \leq \sum_{t=1}^T \sigma_{t-1}(\boldsymbol{x}_t)(\nu_{t^*}^{\frac{1}{2}} + \zeta_T^{\frac{1}{2}}) \leq \sqrt{\frac{2T\gamma_T}{\log(1+\sigma^{-2})}}(\nu_{t^*}^{\frac{1}{2}} + \zeta_T^{\frac{1}{2}})$$ □

In addition to the high probability bound, Lemma C.3 bounds the expected regret:

Lemma C.3. *For EST, the expected regret at time step t is $\mathbb{E}[\tilde{r}_t] = \nu_t^{\frac{1}{2}}\sigma_{t-1}(\boldsymbol{x}_t)$. The expected simple regret is upper bounded by $\mathbb{E}[r_t] = \mathbb{E}[\min_{t\in[1,T]}\tilde{r}_t] \leq \nu_T^{\frac{1}{2}}\sigma_{T-1}(\boldsymbol{x}_T)$.*

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

$$\mathbb{E}[\tilde{r}_t] = \mathbb{E}[Y - f(\boldsymbol{x}_t)|\mathfrak{D}] = m - \mu_{t-1}(\boldsymbol{x}_t) = \nu_t^{\frac{1}{2}}\sigma_{t-1}(\boldsymbol{x}_t).$$

From this, the expected simple regret can be bounded via Jensen's inequality:

$$\mathbb{E}[\min_{1\leq t\leq T}\tilde{r}_t] \leq \min_t \mathbb{E}[\tilde{r}_t] \leq \mathbb{E}[\tilde{r}_T] = \nu_T^{\frac{1}{2}}\sigma_{T-1}(\boldsymbol{x}_T) \qquad \square$$

Theorem 2.4 directly implies a convergence bound on the number of rounds needed to achieve a regret of at most ϵ.

Corollary C.4 (Trial Bound)**.** *Within $T \geq C(\nu_{t^*} + \zeta_T + 2(\nu_{t^*}\zeta_T)^{\frac{1}{2}})\gamma_T/\epsilon^2$ rounds, EST achieves, with probability $1-\delta$, a regret of at most ϵ.*

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

Lemma C.5 (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 $\hat{m}$ and m for the mean field approach.

Corollary C.6. *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 C.6 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, $\hat{m}$ is often still an upper bound on m in practice (illustrations in the supplement).

Given that $\hat{m} \geq m$, Lemma 2.1, 2.4 and Corollary C.4 continue to hold when we use $\hat{m}$ instead of m in the algorithm, so it is not necessary to know m. In contrast, the results above are not necessarily true for any arbitrary θ in GP-PI, an important distinction between GP-PI and GP-EST.

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 β in GP-UCB and θ 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 $\hat{m}$) are observed, then exploitation increases. If we write EST as GP-PI, we see that by Eqn. (2), the estimated $\hat{m}$ always ensures $\theta > \max_{\tau\in[1,T]} y_\tau$, which is known to be advantageous in practice [20]. These analogies likewise suggest that $\theta = \max_{\tau\in[1,T]} y_\tau$ corresponds to a very small β in GP-UCB, and results in very little exploration, offering an explanation for the known shortcomings of this θ.

D Discussion

In this section, we discuss further details and relations of EST.

Setting $\hat{m}$. We showed in Section 2 one way of setting $\hat{m}$ with mean field assumptions. We focused on Eq.(1) and Eq.(2) 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 (e.g., guessing and doubling), and prior knowledge about properties of f such as the range can be taken into account. Our theoretical results continue to hold whenever $\hat{m} \geq m$.

Discretization. We would like to make a few points about discretization, using the noiseless case Eqn. (2) as an example. Eqn. (1) can be analyzed similarly for more general settings. For a Lipschitz continuous function, estimating the probability of $f(\boldsymbol{x}) \leq y, \forall \boldsymbol{x} \in \mathfrak{W}$ (event $E_{\mathfrak{W}}(y)$) on an ϵ-covering of $\mathfrak{W}$ is essentially sufficient.

Since the function f is sampled from GP, we can assume that there exists $a, b > 0$ such that for a fixed $\boldsymbol{x}'$ $\Pr[|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leq L\epsilon, \forall ||\boldsymbol{x} - \boldsymbol{x}'|| \leq \epsilon] \geq 1 - dae^{-L^2/b^2}$ (the same assumption as in [30, Theorem 2]). If $\mathfrak{W}$ is a continuous set, we construct its ϵ-covering $\mathfrak{W}'$ such that $\forall \boldsymbol{x} \in \mathfrak{W}$, $\min_{\boldsymbol{x}' \in \mathfrak{W}'} ||\boldsymbol{x} - \boldsymbol{x}'|| \leq \epsilon$. Then,

$$\Pr[E_{\mathfrak{W}}(y)] \geq \Pr[E_{\mathfrak{W}'}(y - \epsilon L), E_{\mathfrak{W} \setminus \mathfrak{W}'}(y)] \geq \Pr[E_{\mathfrak{W}'}(y - \epsilon L)](1 - |\mathfrak{W}'| dae^{-L^2/b^2})$$

The last step uses Lipschitz continuity and union bound to compute $\Pr[E_{\mathfrak{W} \setminus \mathfrak{W}'}(y) | [E_{\mathfrak{W}'}(y - \epsilon L)]$[2]. 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. Entropy search methods such as ES [12] and PES [13] also compute the event $M_{\boldsymbol{x}}$ ($\boldsymbol{x} = \arg\max_{\boldsymbol{x}' \in \mathfrak{X}} f(\boldsymbol{x}')$). The cited works use expectation propagation to approximate the discrete probability distribution as $\Pr[M_{\boldsymbol{x}}] = \int p(f) \prod_{\boldsymbol{x}' \neq \boldsymbol{x}} \mathbb{1}_{f(\boldsymbol{x}) > f(\boldsymbol{x}')} \, \mathrm{d}f$. The motivation behind those methods is fundamentally different from that of regret-minimizing bandit methods (including EST). The cumulative regret penalizes all queried points, and a method that minimizes regret needs to query enough supposedly good points. ES and PES, 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, i.e., they minimize the entropy of $\Pr[M_{\boldsymbol{x}}]$. They do not pay for any intermediate cost.

Furthermore, our method for computing $\Pr[M_{\boldsymbol{x}}]$ and $\arg\max \Pr[M_{\boldsymbol{x}}]$ is very different from [12, 13]. Our strategy is computationally much more efficient and is meaningful in theory too.

We show an empirical comparison between EST, ES and PES in Section E.5, although the different motivations make a direct comparison difficult.

E Experiments

We next provide more exhaustive experimental results and details on the experiments.

E.1 Approximate m

In the paper, we estimate m to be

[2] Assume $|\mathfrak{W}'| = \frac{c_0}{\epsilon^{c_1}}$ for some $c_0, c_1 > 0$. The right hand side of the inequality above can be written as $\Pr[E_{\mathfrak{W}'}(y - \epsilon b \sqrt{\log \frac{dac_0}{\delta \epsilon^{c_1}}})](1 - \delta)$. We can control ϵ, δ to be small.

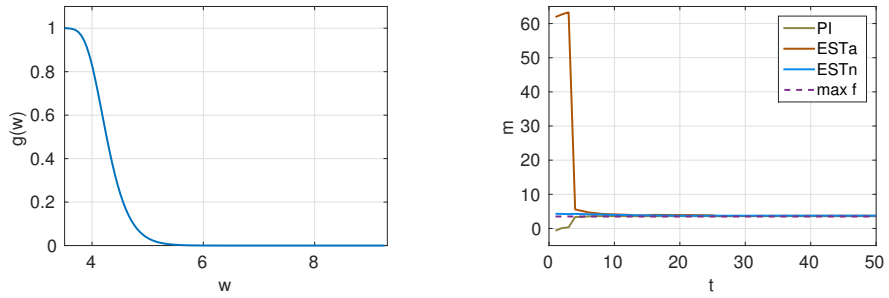

Figure 2: Left: $g(w)$, $w \in [m_0, +\infty)$; right: estimation of m.

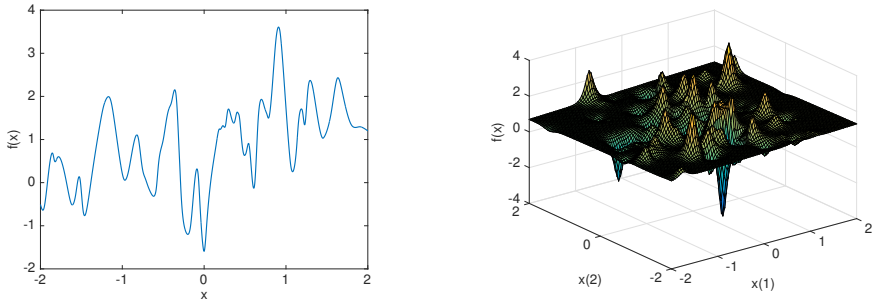

Figure 3: Examples of a function sampled from 1-D GP (left), and a function sampled from 2-D GP (right) with isotropic Matérn kernel and linear mean function. We deliberately create many local optimums to make the problem hard.

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

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 $g(w) = 1 - \prod_{\boldsymbol{x} \in \mathfrak{W}} \Phi\Big(\frac{w - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\Big)$ in Figure 2, 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 2 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, $\theta = \arg\max_{1 \le \tau < t} y_\tau + \epsilon$ is always a lower bound of an ϵ shift over the true maximum of the function.

E.2 Synthetic data

We show the examples of the functions we sampled from GP in Figure 3. 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. For UCB, we follow [30] to set β_t with $\delta = 0.01$. For PI, we use $\epsilon = 0.1$. No hyper-parameter tuning is done in this part of experiments for the covariance function and the mean function. But ϵ in PIand δ in UCBare coarsely tuned for better performance.

Figure 4(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

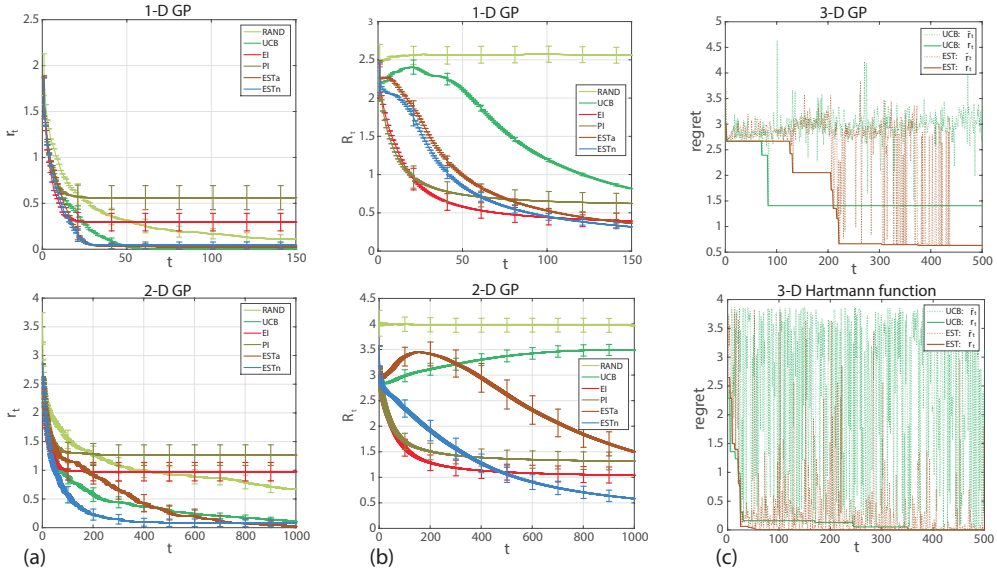

Figure 4: (a) Simple regrets for functions sampled from 1-D GP and 2-D GP over the number of rounds. ESTn quickly minimizes regret. (b) Cumulative regrets for functions sampled from 1-D GP and 2-D GP. ESTn lowers the regret more than PI and EI and faster than UCB. (c) Regrets $\tilde{r}_t$ and simple 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 accumulated regret.

being the fastest. Moreover, the standard deviation of ESTa and ESTn is much lower than that of PI and EI.

Figure 4(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 4(c) plots the cumulative 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 β_t down by a factor of 5), indicating that the scheme of setting β_t in UCB is not always ideal in practice.

E.3 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 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 5. We used 8 configurations of the starting point and fixed the target. 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 $[5\ 9\ 0\ 0]$.

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). We can initialize the trajectory by setting the mid

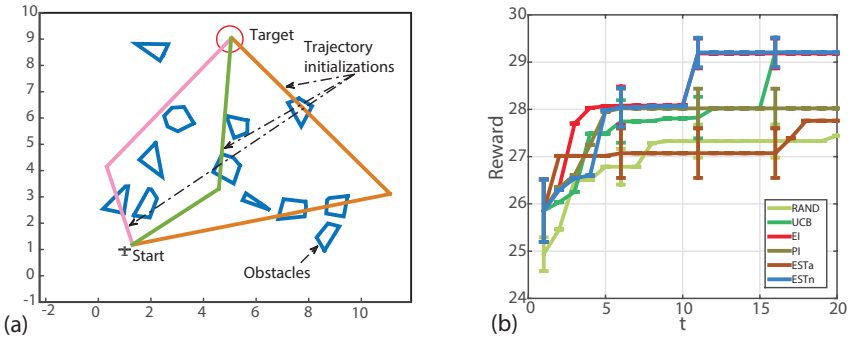

Figure 5: (a) Illustration of trajectory initializations. Trajectory initializations are passed to the non-convex optimization solver to ensure a solution with as few collisions with the obstacles as possible. (b) Maximum rewards up to round t. EI and ESTn perform relatively better than other methods.

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

Table 2: Lowest reward attained in 20 rounds on the airplane problem. The results are averaged over 8 settings.

point of trajectory to be any point falling on the grid of the space (both x axis and y axis have range $[-2, 12]$). We used SNOPT [9] to solve the trajectory optimization problem (SQP), 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 hyper parameters for GP as described in Section E.2 for all the methods we tested. The hyper parameters are tuned every 5 rounds by maximizing the marginal log-likelihood of data.

Figure 5 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 random restarts, indicating a new successful application of Bayesian optimization.

E.4 Parameter tuning for image classification

Our experimental setup follows [37]. Each data set is split into a training, validation (20% of the original training set) and test set. We train a linear SVM on the training set, using the deep features. The SVM parameter (cost in linear SVM) is tuned 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.

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.

We used the same settings of hyper parameters for GP as described in Section E.2 for all the methods we tested. The hyper parameters are tuned every 5 rounds by maximizing the marginal log-likelihood of data.

The following is the experimental setting for the image classification:

In Caltech101 and Caltech256 experiment [8, 10], there are totally 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, all the left images are as test images.

	Caltech101	Caltech256	Indoor67	SUN397	Action40	Event8
[37]	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

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].

Table 4: Classification accuracy on the testing set of the datasets after the model parameter is tuned by ESTa and ESTn. There are improvements over the classification accuracy compared with the previous ones in [37].

	Caltech101	Caltech256	Indoor67	SUN397	Action40	Event8
Imagenet-CNN feature [37]	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 [37]	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

In SUN397 experiment [35], there are totally 108,754 images from 397 scene categories. Images are randomly splitted into training set and test set. The training size is 50 images per category, all the left images are as test images.

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

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

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

Table 4 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].

As shown in [37], deep feature from Convolutional Neural Network (CNN) trained on Places Database has different classification performance betweeen object centric datasets and scene centric datasets, compared with the deep feature from the CNN trained on ImageNet (ImageNet-CNN feature). We further use the deep feature from the ConvNet trained on Places Database (Places-CNN feature) to repeat all the experiments of parameter tuning. The results are shown in Figure 6 and Table 4. All the methods help improve the classification accuracy. On some dataset, the RAND method converges earlier than the other methods, which might be relevant to the property of the visual feature that the deep feature from Places-CNN is not so sensitive to the parameter of the classifier.

E.5 Comparison to entropy search methods

Entropy search methods including ES [12] and PES [13] do not intend to solve GP optimization problems in bandit settings. Their goal is to minimize the entropy of the probability for the event $M_{\boldsymbol{x}}$ ($\boldsymbol{x} = \arg\max_{\boldsymbol{x}' \in \mathfrak{X}} f(\boldsymbol{x}')$). Since both ES and PES only support squared exponential covariance function and zero mean function in their code right now, and it require 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 and 0 mean. Function examples are shown in Figure 7.

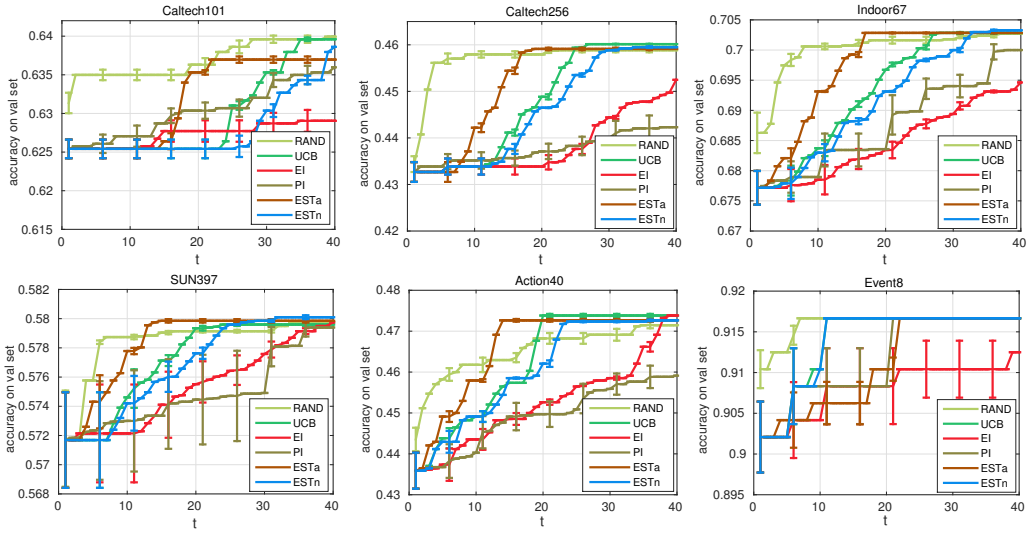

Figure 6: Maximum accuracy on the validation set over the iteration of the optimization. ESTa converges faster than other methods. Experiments are repeated 5 times. The visual feature used here is the deep feature from the Places-CNN.

Table 5: Comparison on 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

We show the results on these synthetic functions in Figure 8,9, and a standard optimization test function, Branin Hoo function, in Figure 10. It is worth noting that ES methods intend to make queries on the most informational inputs. It may seem sometimes these queries also have values that are close to the maximum of the function, e.g. PES in the 2-D GP task and ES in the Branin Hoo task. But in all other tasks, their performance can be arbitrarily bad.

For the results comparing the guesses made by ES methods and the queries made by other methods, we emphasize that these are not fair comparisons, since ES methods do not query their guesses on the blackbox functions.

For the Branin Hoo function, it is interesting to notice that random selection (RAND) actually perform much better than almost all of the other methods. It is not surprising, because those GP optimization methods rely heavily on their prior belief of the GP, and we did not try to optimize the prior belief. One way of circumvent this issue is to marginalize over possible prior parameters and compute an integrated acquisition function [29], which we will not discuss here, since in this paper we focus on the most basic GP optimization technique. One can easily adapt our method to more complicated methods that build on the basic algorithms.

We also compared the running time for all the methods in 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.

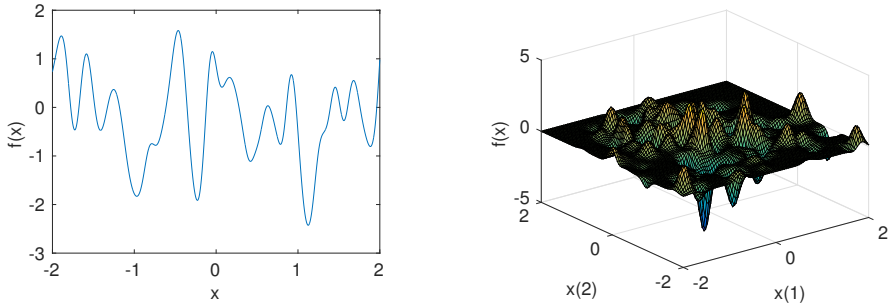

Figure 7: Examples of a function sampled from 1-D GP (left), and a function sampled from 2-D GP (right) with squared exponential kernel and 0 mean functions. These functions can be easier than the ones in Figure 3 since they have fewer local optimums and there are several $\boldsymbol{x}$ that can reach the optimal.

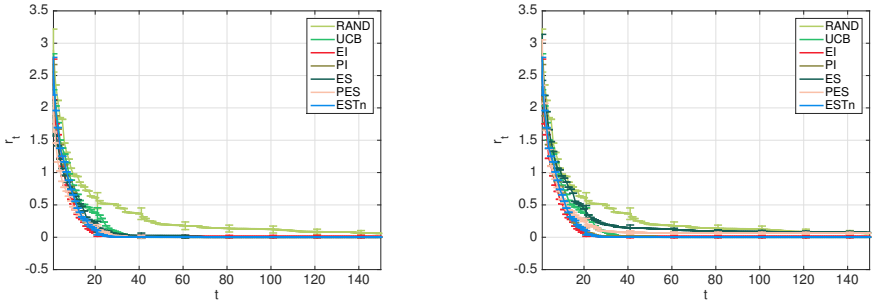

Figure 8: Simple regrets for functions sampled from 1-D GP with squared exponential kernel and 0 mean. Left: ES and PES's regrets are evaluated on their guesses which are never tested on the black-box function when running their algorithm. All methods except RAND tend to behave similarly, possibly because the task is relatively easy (the prior can give a lot of information for the function). Right: ES and PES's regrets are evaluated on the actual inputs they tested on f each round. The inputs tested for ES and PES each round are not designed to minimize the regrets; hence they are not as good as other methods.

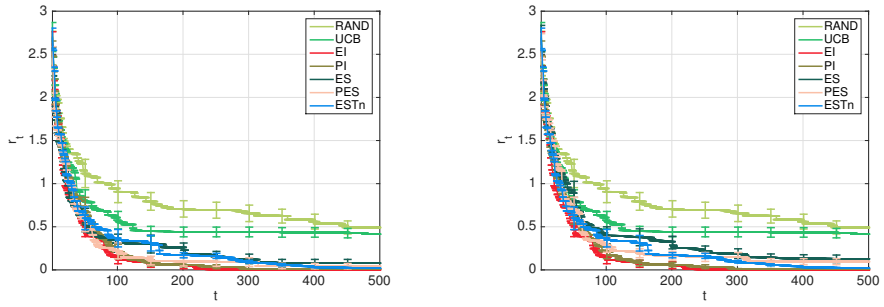

Figure 9: Simple regrets for functions sampled from 2-D GP with squared exponential kernel and 0 mean. Left: ES and PES's regrets are evaluated on their guesses which are never tested on the black-box function when running their algorithm. In general EI and PI perform better than other methods. Similar to the behavior on 1-D GP sample functions, EI and PI did not get stuck in the local optimum for these synthetic test functions, since these are relatively easier tasks than Figure 3. Right: ES and PES's regrets are evaluated on the actual inputs they tested on f each round. They perform slightly worse than EI, PI and EST, but better than UCB and RAND.

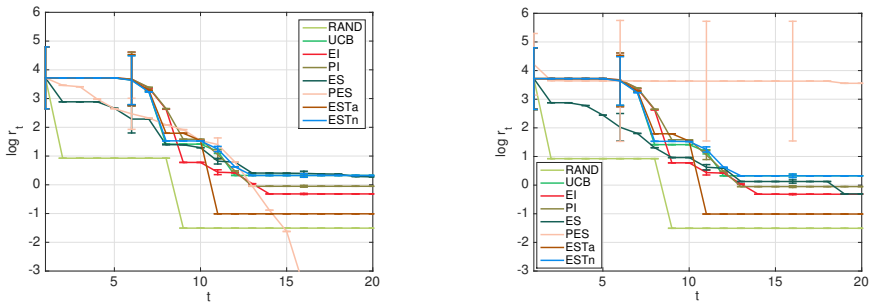

Figure 10: Simple regrets in log scale for Branin Hoo function. UCB/EI/PI/ESTa/ESTn use the isotropic Matérn kernel with parameters $\ell = 0.1, \sigma_f = 1$; ES/PES use the isotropic squared exponential kernel with $\ell = 0.1, \sigma_f = 1$. For this standard test function for optimization, RAND behaves the best. This is possibly because the significant mismatch between the prior of GP and the actual function. Left: regrets for ES/PES are based on their guesses that are not tested during the algorithm. Right: regrets for ES/PES are based on their tested inputs during each round.