Minimizing UCB: a Better Local Search Strategy in Local Bayesian Optimization

Zheyi Fan^{1,2}, **Wenyu Wang**^{3}, **Szu Hui Ng**^{3}[†], **Qingpei Hu**^{1,2}
^{1}Academy of Mathematics and Systems Science, Chinese Academy of Sciences, China
^{2}School of Mathematical Sciences, University of Chinese Academy of Sciences, China
^{3}Department of Industrial Systems Engineering & Management, National University of Singapore, Singapore
^{1,2}`{fanzheyi,qingpeihu}@amss.ac.cn`, ^{3}`{wangwy,isensh}@nus.edu.sg`

Abstract

Local Bayesian optimization is a promising practical approach to solve high dimensional black-box function optimization problem. Among them is the approximated gradient class of methods, which implements a strategy similar to gradient descent. These methods have achieved good experimental results and theoretical guarantees. However, given the distributional properties of the Gaussian processes applied on these methods, there may be potential to further exploit the information of the Gaussian processes to facilitate the Bayesian optimization search. In this work, we develop the relationship between the steps of the gradient descent method and one that minimizes the Upper Confidence Bound (UCB), and show that the latter can be a better strategy than direct gradient descent when a Gaussian process is applied as a surrogate. Through this insight, we propose a new local Bayesian optimization algorithm, MinUCB, which replaces the gradient descent step with minimizing UCB in GIBO [22]. We further show that MinUCB maintains a similar convergence rate with GIBO. We then improve the acquisition function of MinUCB further through a look ahead strategy, and obtain a more efficient algorithm LA-MinUCB. We apply our algorithms on different synthetic and real-world functions, and the results show the effectiveness of our method. Our algorithms also illustrate improvements on local search strategies from an upper bound perspective in Bayesian optimization, and provides a new direction for future algorithm design.

1 Introduction

Bayesian Optimization [8] is one of the most well-known black box function optimization methods, where objectives can be extremely expensive to evaluate, noisy, and multimodal. The high efficiency of Bayesian Optimization in finding global optima leads to the widespread application in various research fields, such as hyperparameter tuning [10, 11, 28], neural architecture search [26], chemical experiment design [12], reinforcement learning [31], aerospace engineering [16]. However, the performance of Bayesian optimization is limited by the input dimension d, as the theoretical regret bound grows exponentially with input dimension [25]. This difficulty hinders the application of Bayesian optimization when the actual dimension of problem is rather high [8].

There are various methods that have been proposed to handle this difficult task, including works that rely on some assumptions on the model structure, such as the assumption that the majority of the variables have no effect [6] or the kernel satisfies an additive structure [9]. Local Bayesian optimization methods, which focus on finding a local optima (instead of the global one), have also been a popular (and less restrictive) compromise to manage the curse of dimensionality. Representative

[†] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

methods of these include those based on local trust region methods [7], local latent space [20], and approximated gradient methods [22, 23, 36]. Among them the approximate gradient method has demonstrated strong performance in practical applications compared with other methods. The approximate gradient method can be described as a two-stage algorithm, which loops through the following two processes: first sample points to decrease the uncertainty of the local area according to a local exploration acquisition function, and then moving to the next point with a trustworthy high reward through a local exploitation acquisition function. Müller et al. [22] applied the idea of gradient descent and first proposed the GIBO algorithm, which was designed to alternate between sampling points to minimize the posterior variance of the gradient at a given location, and then moving in the direction of the expected gradient. Nguyen et al. [23] proposed MPD, which improved the local exploitation acquisition function through defining the descent direction by maximizing the probability of descent, and designed corresponding local exploration acquisition function to match this strategy. Wu et al. [36] further provided the detailed proof on the local convergence of GIBO with a polynomial convergence rate, for both the noiseless and noisy cases.

Although the approximated gradient method has been shown to be practical in dealing with high-dimensional problems, there may be still some room to potentially improve on the current methods. We motivate the ideas for improvement with the following two questions.

1) We observe that GIBO only utilizes the posterior distributions of the gradient at a point, which will ignore most of the information provided by Gaussian process surrogate in the region, which may lead to an inefficient descent. MPD attempts to make better use of Gaussian processes by performing multi-step descent, but this strategy can exhibits numerical instability and may lead to suboptimal performance of the algorithm (as seen in Section 8). This motivates us to think: is there a better local exploitation acquisition function that can ensure the algorithm fully utilizes Gaussian process information, and also guarantee the convergence to local optima points?

2) Do these acquisition functions necessarily need to depend on accurate gradient estimates at a point, or are there other acquisition functions that can improve the efficiency and still ensure local convergence?

In this paper, we attempt to answer the above two questions through our two new local Bayesian optimization algorithms. To address the first question, we first develop the relationship between the step of the gradient descent method and minimizing the Upper Confidence Bound (UCB). When the Gaussian process is applied as the surrogate model, minimizing the UCB can usually achieve a point with a higher reward than simply doing gradient descent. Motivated by this idea, we propose our first algorithm, Local Bayesian Optimization through Minimizing UCB (MinUCB), which replaces gradient descent step with a step that minimizes the UCB in the GIBO algorithm. We show that MinUCB will also converge to local optima with a similar convergence rate as GIBO. This discovery is also meaningful as it opens up possibilities for new designs on local Bayesian optimization algorithms. In this work we further apply the look ahead strategy to construct the local exploration acquisition function that is more compatible with minimizing the UCB, and propose our second algorithm, Look Ahead Bayesian Optimization through Minimizing UCB (LA-MinUCB). This algorithm is shown to be one step Bayesian optimal, and address our second question as a local efficient BO acquisition function does not require the additional step and accuracy of a gradient estimate. We implement extensive experiments to demonstrate the performance of our algorithms under different settings. The experimental results illustrate that our algorithms have better performance compared to other methods across many synthetic and real-world functions. We summarize our contributions as follows:

- We develop the relationship between the gradient descent step and minimizing the UCB, and show that minimizing UCB is more efficient when the Gaussian process is the underlying surrogate.
- We show that minimizing UCB is an efficient and accurate objective for local exploitation and propose MinUCB.
- We improve the local exploration acquisition function of MinUCB and obtain a more efficient local Bayesian optimization algorithm LA-MinUCB.
- We apply different synthetic and real-world function on our algorithm, and the results show the effectiveness of our methods.

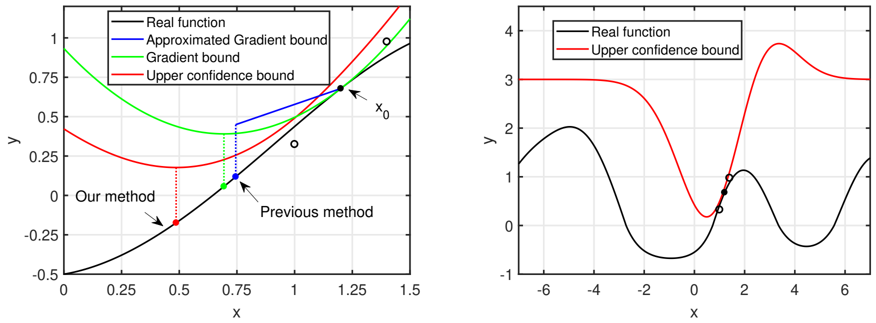

Figure 1: This function f is sampled from $GP(0, k(x, x^{'}))$, where $k(x, x^{'}) = exp(-\frac{1}{4}(x - x^{'})^2)$, with standard derivation of white noise $\sigma = 0.05$. The dataset contains 2 points, which is marked as black hollow circle. We attempt to search the next point from $\mathbf{x}_0$. The left figure shows that UCB bound is much tighter than other two gradient based bounds, and the minimum points of UCB has the best performance. This shows that minimizing UCB in this example can achieve a much better move to lower point than the gradient descent approach. The right figure illustrates UCB across the design space. Here we see that it is small only near the sampled point, and increases as it moves further away, indicating that minimizing UCB can be viewed as local strategy.

2 Literature Review

High dimensional optimization is a growing research area, where many different methods have been proposed to solve this problem, including Bayesian optimization methods. Among the BO methods, the most widely studied approach focuses on structured Gaussian processes, which imposes additional assumptions, such as low dimensional active spaces or additive spaces, on kernel function or data structure. This includes the Additive and ANOVA Models [4], in which the kernel of Gaussian process is defined through the summation of univariate kernels. This essentially decompose the original Gaussian process into a sum of low dimensional Gaussian processes, which facilitates computations and convergence [9, 37, 13]. Another approach along these lines assumes that the objective function depends only on low dimensional subspace, and examples include hyperparameter optimization for neural networks [3]. Most of these methods suppose a subspace is a linear subspace of the original Euclidean space. Wang et al.[34] apply random linear embeddings to approximate this subspace and propose REMBO. Letham et al. [18] improve on this and tries to fix over exploration of boundary and distortions in embedding through adaptive linear embeddings [18]. Other methods to learn the subspace structure include low-dimension matrix recovery [5] and nonlinear embeddings [21].

As mentioned in the Section 1, another line of research to address the computational and high dimensional challenges is to compromise and focus the BO to be more local, limiting the search region to facilitate computational feasibility and efficiency. Typical approaches along these lines look at incorporating the information about the local optimum [1], restricting the problem to a sequence of iteratively chosen one-dimensional sub-problems [14, 15], using trust regions [7, 32, 20], which expands and shrinks the size of its trust regions with information in each iteration, and approximated gradient methods [22, 23, 36]. Among them, the gradient based method MPD [23], proposed by Nguyen et al. has demonstrated very strong performance compared with other local methods, including gradient based method GIBO [22], trust region method Turbo [7] and Augmented Random Search (ARS) [19]. Given the promising performance of approximate gradient methods, we leverage and extend the works of GIBO and MPD, to provide a simplier and more general local BO approach that can perform as well.

3 Preliminaries

3.1 Bayesian Optimization

In this paper, we focus on the problem of minimizing a black-box function $f(\cdot)$:

$$\min_{\mathbf{x} \in \mathbb{X}} f(\mathbf{x})$$

We assume that no higher order information can be obtained from the oracle (Zeroth-Order-Oracle), where only $i.i.d$ noisy function evaluations $y = f(\mathbf{x}) + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ can be observed. Here we focus on applying Bayesian optimization to solve the problem. BO is a surrogate based optimization approach where the objective function is typically modeled with a Gaussian process, and an acquisition function is constructed to sequentially determine next evaluation points in the optimization process.

3.2 Gaussian process and its derivatives

Currently Gaussian process (GP) is one of the most widely used surrogate model today as it has nice analytical form and is flexible to capture various functional forms. A $GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^{'}))$ is specified by its mean function $m(\cdot)$ and kernel function $k(\cdot, \cdot)$. Without loss of generality, we assume the mean function $m(\mathbf{x}) \equiv 0$. Suppose f is sampled from the $GP(0, k(\mathbf{x}, \mathbf{x}^{'}))$, and we already have a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, and set $\mathbf{X}_\mathcal{D} = [\mathbf{x}_1, ..., \mathbf{x}_n]$, $\mathbf{y}_\mathcal{D} = [y_1, ..., y_n]^T$, then the posterior over f is also a $GP(\mu_\mathcal{D}(\mathbf{x}), k_\mathcal{D}(\mathbf{x}, \mathbf{x}^{'}))$, where

$$\begin{aligned}
\mu_\mathcal{D}(\mathbf{x}) &= k(\mathbf{x}, \mathbf{X}_\mathcal{D})(k(\mathbf{X}_\mathcal{D}, \mathbf{X}_\mathcal{D}) + \sigma^2 I)^{-1}\mathbf{y}_\mathcal{D} \\
k_\mathcal{D}(\mathbf{x}, \mathbf{x}^{'}) &= k(\mathbf{x}, \mathbf{x}^{'}) - k(\mathbf{x}, \mathbf{X}_\mathcal{D})(k(\mathbf{X}_\mathcal{D}, \mathbf{X}_\mathcal{D}) + \sigma^2 I)^{-1}k(\mathbf{X}_\mathcal{D}, \mathbf{x}^{'}) \\
\sigma^2_\mathcal{D}(\mathbf{x}) &= k_\mathcal{D}(\mathbf{x}, \mathbf{x})
\end{aligned}$$

where $k(\mathbf{x}, \mathbf{X}_\mathcal{D}) = [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_n)]$, $k(\mathbf{X}_\mathcal{D}, \mathbf{x}) = k(\mathbf{x}, \mathbf{X}_\mathcal{D})^T$, and $k(\mathbf{X}_\mathcal{D}, \mathbf{X}_\mathcal{D})$ is the positive definite kernel matrix $[k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j \le n}$.

It should be noted that if the kernel function k is differentiable, then the derivative of f is also a GP. Given the dataset $\mathcal{D}$, The posterior of f satisfies:

$$\nabla f | \mathcal{D} \sim GP(\nabla \mu_\mathcal{D}(\cdot), \nabla k_\mathcal{D}(\cdot, \cdot)\nabla^T)$$

In this work we only consider the noisy case, i.e. the standard derivation of white noise $\sigma > 0$.

4 The relationship between gradient descent and minimizing UCB

We first review the traditional gradient descent approach. This approach is fundamentally based on the assumption of the smoothness of the function:

Definition 1. *(Smoothness) A function f is L-smooth if and only if for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{X}$, we have*

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\|_2 \le L\|\mathbf{x}_1 - \mathbf{x}_2\|_2$$

Suppose the initial point is $\mathbf{x}_0$, and function $f(\cdot)$ is L-smooth, then we will have the following inequality:

$$f(\mathbf{x}) \le f(\mathbf{x}_0) + \langle \nabla f(\mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2 \tag{1}$$

This provides a quadratic upper bound on f, and the minimum value of this upper bound is taken at $\mathbf{x} = \mathbf{x}_0 - \eta \nabla f(\mathbf{x}_0)$. In the gradient descent approach, the minimum value of this upper bound is taken as the descent step, where $\eta = \frac{1}{L}$ is used as the step size in traditional gradient descent analysis. In this view gradient descent can be treated as selecting the minimum point of this quadratic upper bound Eq. (1).

As direct gradient information is not observable in practice, the upper bound Eq. (1) cannot be obtained, and approximate gradient methods instead attempt to derive a looser upper bound based on Eq. (1). In GIBO [22, 36], they replace the gradient $\nabla f(\mathbf{x}_0)$ with the derivative of the Gaussian process $\nabla \mu_\mathcal{D}(\mathbf{x}_0)$, and apply the strategy $\mathbf{x} = \mathbf{x}_0 - \eta \nabla \mu_\mathcal{D}(\mathbf{x}_0)$ when the variance of this gradient is small enough. According to the proof of Lemma 15 in [36], GIBO is also a result of optimizing a different upper bound of $f(\cdot)$. If $\mathbf{x}$ is chosen to be the form of $\mathbf{x} = \mathbf{x}_0 - \eta \nabla \mu_\mathcal{D}(\mathbf{x}_0)$ and $\eta \le \frac{1}{L}$, this upper bound is:

$$f(\mathbf{x}) \le f(\mathbf{x}_0) - \frac{1}{2}\eta\|\nabla f(\mathbf{x}_0)\|_2^2 + \frac{1}{2}\eta\|\nabla \mu_\mathcal{D}(\mathbf{x}_0) - \nabla f(\mathbf{x}_0)\|_2^2 \tag{2}$$

If the approximation error of gradient $\|\nabla \mu_\mathcal{D}(\mathbf{x}_0) - \nabla f(\mathbf{x}_0)\|_2$ is small enough , then the optimal η is chosen as $\frac{1}{L}$. This upper bound Eq.(2) is actually the local exploitation acquisition function of GIBO.

Algorithm 1: MinUCB: Local Bayesian Optimization through Minimizing UCB

```
Input: A black-box function $f$, and initial point $\mathbf{x}_1$
for $t = 1, 2, ..., T$ do
    $\mathbf{X}_1 = [\mathbf{x}_t^T, ..., \mathbf{x}_t^T]^T \in \mathbb{R}^{b_t^{(1)} \times d}$        #Resample multiple times on $\mathbf{x}_t$
    $\mathbf{X}_2 = \arg\min_{\mathbf{Z}} \alpha_{\text{trace}}(\mathbf{x}_t, \mathbf{Z})$ where $\mathbf{Z} \in \mathbb{R}^{b_t^{(2)} \times d}$        #Local exploration (sampling)
    $\mathbf{X} = [\mathbf{X}_1^T, \mathbf{X}_2^T]^T$
    evaluate the black-box function $f$ on $\mathbf{X}$, obtaining noisy measurements $\mathbf{y}$
    $\mathcal{D}_t = \mathcal{D}_{t-1} \cup (\mathbf{X}, \mathbf{y})$
    $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} \mu_{\mathcal{D}_t}(\mathbf{x}) + \beta_t \sigma_{\mathcal{D}_t}(\mathbf{x})$        #Local exploitation (step move)
end
```

Although these two bounds have intuitive application in the gradient descent, they do have some limitations. The first is the obtainment of the L-smooth coefficient. Although it is possible to estimate the L-smooth coefficient through the Gaussian process, this estimation is expensive as it needs many samples, especially in high dimensional cases. The second is that these two bounds are relatively loose, and the minimum points of these two bounds tend to be too close to $\mathbf{x}_0$. Upper bound Eq. (1) is quadratic and increase very fast when the point $\mathbf{x}$ is far from $\mathbf{x}_0$, and upper bound Eq. (2) only allows the stepsize η to be less than $\frac{1}{L}$. When L is unknown and we have to give it a large estimate to ensure convergence (usually in real case), the above phenomenon becomes more severe. Taking this view of gradient descent approaches as moving along the minimum of an upper bound of $f(\cdot)$, then leads us to explore if it is possible to discover some tighter upper bounds, where the minimum point is lower than that in Eq. (1) and Eq. (2). This can lead us to find a point with a possible higher reward.

A commonly used concept in Bayesian Optimzation is the upper confiedence bound (UCB), which is defined as followed:

$$\text{UCB}(\mathbf{x}) = \mu_{\mathcal{D}}(\mathbf{x}) + \beta\sigma_{\mathcal{D}}(\mathbf{x}) \tag{3}$$

Previous work mainly focused on maximizing UCB to find the maximum value of a function [29]. However, it should be noted that UCB is also a natural bound for function $f(\cdot)$. UCB fully utilizes the posterior distribution of $f(\cdot)$, and give every point a probabilistic bound depending on the coefficient β. The standard deviation term $\sigma_{\mathcal{D}}(\mathbf{x})$ has an upper bound and will not grow faster than the quadratic function, which means the UCB will not change drastically. This indicates that if we select $\mathbf{x}$ to be

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mu_{\mathcal{D}}(\mathbf{x}) + \beta\sigma_{\mathcal{D}}(\mathbf{x}) \tag{4}$$

Then the function value $f(\mathbf{x}^*)$ more likely to have a smaller value than the points obtained through optimizing the upper bound Eq. (1) and Eq. (2). This is because the points obtained by minimizing UCB can be further away from the initial point compared to simply gradient descent.

Fig (1) shows a simple 1-dimension illustrative example. In this example, we sample a function f from $GP(0, k(x, x^{'}))$, where $k(x, x^{'}) = exp(-\frac{1}{4}(x - x^{'})^2)$, and the standard deviation of white noise $\sigma = 0.05$. Here we illustrate the search of the next point from x_0 based on the upper bound perspective of three methods. Suppose we have already sampled two points, which are marked as black hollow circles. These two data points are selected through Central Finite Difference Approximations [27], which is aimed to better estimate the gradient of $\nabla f(\mathbf{x}_0)$. In the left figure, the green line represent the real quadratic upper bound Eq. (1) at $\mathbf{x}_0$, and the blue line is calculated through Eq. (2). The coefficient β in UCB bound Eq. (3) is set as 3, which means that for any point $\mathbf{x}$, the probability of $f(\mathbf{x}) < UCB(\mathbf{x})$ is close to 99.9%. It can be seen from the left figure in Fig (1) that the UCB bound is much tighter than other two bounds, and the minimum point of the UCB bound has a much lower function value than the gradient based method. The right figure plots the UCB across the input space, and we see that the UCB changes relatively slowly and will not reach infinity. Further we observe that UCB is only small near the sampled point, indicating that minimizing the UCB can be viewed as a local strategy. This simple example illustrates that with a Gaussian process function, minimizing UCB can achieve a better point than the gradient methods, as UCB efficiently utilizes the information from Gaussian process. Based on these insights, we propose two new local Bayesian optimization algorithms, and demonstrate their performances in several numerical examples.

5 Local Bayesian Optimization through Minimizing UCB

The analysis the above section provides us with an important idea, that if we replace the gradient descent step with a step that minimizes a tighter upper bound such as UCB, we may be able to achieve a better result in local optimization. Our first algorithm, Local Bayesian Optimization through Minimizing UCB (MinUCB) (Algorithm 1), is developed with this idea, and we show that minimizing the UCB for the step move is an efficient objective for the local exploitation, that can guarantee the convergence with an appropriate local exploration acquisition function.

MinUCB can be viewed as a modified version of GIBO [22, 36] (we list GIBO algorithm in Appendix A for reference). In our approach, we adopt the same local exploration acquisition function to sample points as in GIBO (to keep that constant) (line 4 in Algorithm 1), and only set the objective of local exploitation acquisition function that drives the step move to minimizing the UCB, instead of gradient descent step $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla \mu_{\mathcal{D}_t}(\mathbf{x}_t)$, as shown in line 8 in Algorithm 1. We first introduce some notations here for better illustration of our algorithm. We define $k_{\mathcal{D}\cup\mathbf{Z}}(\mathbf{x}_t, \mathbf{x}_t) = k_{\mathcal{D}}(\mathbf{x}_t, \mathbf{x}_t) - k_{\mathcal{D}}(\mathbf{x}_t, \mathbf{Z})(k_{\mathcal{D}}(\mathbf{Z}, \mathbf{Z}) + \sigma^2 I)^{-1} k_{\mathcal{D}}(\mathbf{Z}, \mathbf{x}_t)$, which is exactly the posterior variance of $f(\mathbf{x}_t)$ conditioned on the dataset $\mathcal{D}$ and a new input $\mathbf{Z}$. Because the estimation of variance does not require $\mathbf{y}$, we have omitted the symbol here. The local exploration acquisition function for sampling is defined on this posterior variance:

$$\alpha_{\text{trace}}(\mathbf{x}_t, \mathbf{Z}) = \mathbf{tr}(\nabla k_{\mathcal{D}_{t-1}\cup\mathbf{Z}}(\mathbf{x}_t, \mathbf{x}_t) \nabla^T) \tag{5}$$

which is the trace of the posterior covariance matrix of the $\nabla f(\mathbf{x}_t)$ conditioned on the dataset and input. This trace quantifies the uncertainty of gradient $\nabla f(\mathbf{x}_t)$. With a large batch size $b_t^{(2)}$, minimizing this trace will result in a lower uncertainty on the estimation of gradient. Although UCB doesn't involve gradient descent, we keep this step constant and argue that the candidates selected through this local exploration acquisition function will still efficiently decrease the uncertainty on this local area, which will benefit the local exploitation move when minimizing the UCB.

In the local exploitation part, the β_t controls the search area for each step. The larger the β_t, the closer $\mathbf{x}_{t+1}$ will be to the existing data point. Minimizing the UCB can bring performance improvements, and we show later in Section 6 (Theorem 2) that MinUCB will have a similar convergence rate as GIBO with carefully selection on coefficients β_t and batch size $b_t^{(1)}$, $b_t^{(2)}$. The results of MinUCB can provide inspiration on the design for more efficient local Bayesian optimization algorithm, as shown in Section 7.

It should be noted that in MinUCB we resample multiple times on the local exploitation result $\mathbf{x}_t$ with a batch size $b_t^{(1)}$ (line 3 in Algorithm 1). This step is mainly added to ensure the theoretical convergence of the algorithm. This step will typically be only a very small proportion of sampling points, especially in the high dimensional case.

6 Convergence Analysis of MinUCB

In this section, we establish a convergence analysis of MinUCB to demonstrate the effectiveness of using minimizing UCB as the objective of local exploitation. We prove that MinUCB has a polynomial convergence rate, and this rate also exhibits a polynomial relationship with the input dimension, indicating that MinUCB performs very well in high-dimensional case. For the whole convergence proof of MinUCB, please refer to Appendix B.1-B.4. In our convergence analysis, we set a mild assumption on kernel function k:

Assumption 1. *The kernel $k(\cdot, \cdot)$ is stationary, four times continuously differentiable, strictly positive definite, and bounded:* $\max_{\mathbf{x}\in\mathbb{X}} k(\mathbf{x}, \mathbf{x}) \leq 1$

Many common kernels such as RBF kernel and Matérn kernel with $\gamma > 2$ will satisfy this assumption. We also need the definition domain of Gaussian process is bounded:

Assumption 2. *The Gaussian process $f(\mathbf{x})$ is defined on a bounded closed set $\mathbb{X}$, i.e. there exist a constant $r > 0$ that $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{X}$, $\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq r$.*

The next we borrow the definition of Error function from Wu et al. [36]. The Error function measures the maximum reduction of uncertainty about the gradient estimation at $\mathbf{x} = 0$ when there are b data points $\mathbf{Z}$ without any extra dataset:

Definition 2. *(Error function) Given input dimensionality d, kernel k and noise standard deviation σ, we define the following error function:*

$$E_{d,k,\sigma}(b) = \inf_{\mathcal{Z}\in\mathbb{R}^{b\times d}} \boldsymbol{tr}(\nabla k(\mathbf{0},\mathbf{0})\nabla^T - \nabla k(\mathbf{0},\mathbf{Z})(k(\mathbf{Z},\mathbf{Z}) + \sigma^2 I)^{-1} k(\mathbf{0},\mathbf{Z})\nabla^T) \tag{6}$$

Based on the above assumptions, we develop the convergence theory for MinUCB:

Theorem 1. *Suppose f is sampled from a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot,\cdot)$, and $k(\cdot,\cdot)$ satisfies Assumption 1. Then after t iterations of MinUCB algorithm, with the batch size $b_t^{(1)}$ and $b_t^{(2)}$, it satisfies that*

$$\begin{aligned}\min_{T/2\le t\le T} \|\nabla f(\mathbf{x}_t)\|_2 \le \frac{1}{\sqrt{\eta_T}} &\sqrt{\frac{8}{T}\sum_{t=1}^{T} \frac{\tilde{\beta}_t \sigma}{\sqrt{b_t^{(1)}}} + \frac{8\pi}{T}\sum_{t=1}^{T} \tilde{\beta}_t^2 \eta_t E_{d,k,\sigma}(b_t^{(2)}) + O(\frac{1}{T}\log\frac{1}{\delta})} \\ &+ \frac{1}{\sqrt{\eta_T}} \tilde{\beta}_{T/2} \sqrt{\frac{\pi}{2}} \sqrt{E_{d,k,\sigma}(b_{T/2}^{(2)})}\end{aligned}$$

where $\tilde{\beta}_t$ and η_t are both decreasing sequence. They satisfies $\tilde{\beta}_t = O(\beta_t)$ and $\frac{1}{\eta_t} = O(d\sqrt{\log\frac{t^2 d^2}{\delta}} + d^{\frac{3}{2}})$, and $\beta_t = \sqrt{2\log\frac{\pi^2 t^2}{\delta}}$

Proof Outline of Theorem 1:

- From the definition of $\mathbf{x}_{t+1}$,we can obtain $f(\mathbf{x}_{t+1}) \le \min_{x\in\mathbb{R}^d} \mu_{\mathcal{D}_t}(\mathbf{x}) + \beta_t \sigma_{\mathcal{D}_t}(\mathbf{x}) \le \mu_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}) + \beta_t \sigma_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1})$, where $\hat{\mathbf{x}}_{t+1}$ is a pseudo gradient descent step. $\hat{\mathbf{x}}_{t+1} = \mathbf{x}_t - \eta_t \nabla \mu_{\mathcal{D}_t}(\mathbf{x}_t)$. The probability of the above inequality being true can be controlled by β_t.
- We try to build the relationship of $\mu_{\mathcal{D}_t}(\mathbf{x}_t) + \beta_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t)$ and $\mu_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}) + \beta_t \sigma_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1})$. We apply the local properties of the mean function $\mu_{\mathcal{D}_t}(\mathbf{x})$ and standard deviation function $\sigma_{\mathcal{D}_t}(\mathbf{x})$ (proved in Appendix B.1 and B.2), which is:

$$\begin{aligned}\mu_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}) &\le \mu_{\mathcal{D}_t}(\mathbf{x}_t) - \eta_t \|\nabla \mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 + \frac{L_{\mu_t}}{2}\eta_t^2 \|\nabla \mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 \\ \sigma_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}) &\le c_1 \sigma_{\mathcal{D}_t}(\mathbf{x}_t) + error(\|\nabla \mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2)\end{aligned}$$

- We bring the above relationship into the initial inequality, and use the triangle inequality to establish the relationship between $\nabla \mu_{\mathcal{D}_t}(\mathbf{x}_t)$ and $\nabla f(\mathbf{x}_t)$. After a few more simple steps of derivation, the final result can be obtained.

It can be seen that the convergence of gradient $\|\nabla f(\mathbf{x}_t)\|_2$ heavily depend on the batch size $b_t^{(1)}$ and $b_t^{(2)}$. The large batch size will accelerate the convergence, with the additional cost of sampling. The upper bound of $E_{d,k,\sigma}(b)$ and its theoretical property can be referred in Wu et al. [36], and we also list them at Appendix B.3, lemma 6-9. Thus if we combine the above result, the convergence rate of MinUCB is obtained:

Theorem 2. *Suppose f is sampled from a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot,\cdot)$, then if the kernel is RBF kernel or Matérn kernel with $\gamma = 2.5$, and satisfy $\beta_t = \sqrt{2\log\frac{\pi^2 t^2}{\delta}}$, batch size*

$$b_t^{(1)} = \begin{cases}\log^2 t \\ t \\ t^2\end{cases} \quad \text{and} \quad b_t^{(2)} = \begin{cases} d\log^2 t \\ dt \\ dt^2\end{cases}$$

Then MinUCB will achieve the convergence rate of

$$\min_{T/2\le i\le T} \|\nabla f(x_i)\|_2^2 \le \begin{cases} O(\sigma d^{\frac{3}{2}} T^{-1} \log^{\frac{3}{2}} \frac{d^2 T^2}{\delta}) + O(\sigma d^2) \\ O(\sigma d^2 T^{-\frac{1}{2}} \log^{\frac{5}{2}} \frac{d^2 T^2}{\delta}) = O(\sigma d^{\frac{9}{4}} n^{-\frac{1}{4}} \log^{\frac{5}{2}} \frac{dn}{\delta}) \\ O(\sigma d^2 T^{-1} \log^{\frac{5}{2}} \frac{d^2 T^2}{\delta})) = O(\sigma d^{\frac{7}{3}} n^{-\frac{1}{3}} \log^{\frac{5}{2}} \frac{dn}{\delta}))\end{cases}$$

where n is the number of samples. Our method achieves similar results to Wu et al. [36], except for some logarithmic term and the order of d. The dimension d in our boundary is larger than that in

Algorithm 2: LA-MinUCB: Look Ahead Bayesian Optimization through Minimizing UCB

```
Input: A black-box function $f$.
for $t = 1, 2, ..., T$ do
    $\mathbf{X} = \arg\min_{\mathbf{Z}} \mathbb{E}_{\mathbf{y_Z}} \min_{\mathbf{x}} \text{UCB}(\mathbf{x}, \mathcal{D}_{t-1}, \mathbf{Z}, \mathbf{y_Z})$ where $\mathbf{Z} \in \mathbb{R}^{b_t \times d}$   #Local exploration
    evaluate the black-box function $f$ on $\mathbf{X}$, obtaining noisy measurements $\mathbf{y}$
    $\mathcal{D}_t = \mathcal{D}_{t-1} \cup (\mathbf{X}, \mathbf{y})$
    $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} \mu_{\mathcal{D}_t}(\mathbf{x}) + \beta_t \sigma_{\mathcal{D}_t}(\mathbf{x})$   #Local exploitation via minimizing UCB
    $y_{t+1} = f(\mathbf{x}_{t+1}) + \varepsilon_t$
    $\mathcal{D}_t = \mathcal{D}_t \cup (\mathbf{x}_{t+1}, y_{t+1})$
end
$\mathbf{x}_T = \arg\min_{\mathbf{x}} \mu_{\mathcal{D}_T}(\mathbf{x}) + \beta_T \sigma_{\mathcal{D}_T}(\mathbf{x})$
```

Wu's work, which is because we also take the upper bound of the L-smooth coefficient of Gaussian process into consideration, while this upper bound also increases at the polynomial rate with the data dimension d, as seen in Theorem 3 in Appendix B.1. According to Theorem 2, we need to iteratively increase the UCB coefficient β_t and batch size $b_t^{(1)}$, $b_t^{(2)}$ to guarantee the convergence of MinUCB. This phenomenon can be explained that, when the algorithm approaches the local optima, the area around local optima will usually be flatter than other areas (as the gradient is near 0). The algorithm needs more detailed local exploration to ensure a better descent. The polynomial convergence rate demonstrates that our local exploitation strategy, minimizing UCB, is accurate and powerful. It can ensure the accuracy of the local search and fully utilize the information of Gaussian processes.

7 Look Ahead Bayesian Optimization through Minimizing UCB

Our proposed MinUCB enjoy good theoretical properties and provide an alternative idea of minimizing the UCB as a good way to progress the local search under a Gaussian process surrogate. However, there are still improvements that can be made to the local exploration in MinUCB. Specifically, UCB itself does not require any gradient information, and the local exploration in MinUCB still focuses on learning the information at a single current point. This under utilizes the Gaussian process surrogate, and the potential information in the local region. Based on the above, we focus here on selecting a better local exploration acquisition function for minimizing UCB, which can help to accelerate the local Bayesian optimization.

In this section we apply a look ahead strategy. The motivation here is to obtain desired candidates to improve the UCB bound, and help the next local exploitation achieve better results under an expectation view. With this idea we propose our second algorithm, Look Ahead Bayesian Optimization through Minimizing UCB (LA-MinUCB) (Algorithm 2).

Suppose we have the input $\mathbf{Z}$ and their labels $\mathbf{y}_Z$ (this part is unknown before sampled), we define $\mathcal{D}_Z$ as $\mathcal{D}_Z = \{(\mathbf{z}_i, y_i)\}$, $i = 1, ..., b_t$ as the dataset formed through $\mathbf{Z}$ and $\mathbf{y}_Z$. Then we define

$$\text{UCB}(\mathbf{x}, \mathcal{D}_{t-1}, \mathbf{Z}, \mathbf{y_Z}) = \mu_{\mathcal{D}_{t-1} \cup \mathcal{D}_Z}(\mathbf{x}) + \beta_t \sigma_{\mathcal{D}_{t-1} \cup \mathcal{D}_Z}(\mathbf{x})$$

This is the upper confidence bound when we already have the dataset $\mathcal{D}_{t-1} \cup \mathcal{D}_Z$, and we want to find the input $\mathbf{Z}$ to minimize the minimum point of UCB: $\min_{\mathbf{x}} \text{UCB}(\mathbf{x}, \mathcal{D}_{t-1}, \mathbf{Z}, \mathbf{y_Z})$. However, as the label $\mathbf{y}_Z$ is unknown, we can only choose to optimize it through its expectation. We adopt this look ahead predictive as the local exploration acquisition function in LA-MinUCB:

$$\mathbf{X} = \arg\min_{\mathbf{Z}} \mathbb{E}_{\mathbf{y_Z}} \min_{\mathbf{x}} \text{UCB}(\mathbf{x}, \mathcal{D}_{t-1}, \mathbf{Z}, \mathbf{y_Z}) \tag{7}$$

Although the local exploration in LA-MinUCB does not need to be specified around a certain point, it is still necessary to have an local exploitation step. Local exploitation step may find a point with current best reward, and will provide a better foundation for subsequent local exploration.

LA-MinUCB has a similar structure with traditional Knowledge Gradient [8] except for the standard derivation term $\sigma_{\mathcal{D}_t}(\mathbf{x})$. This standard derivation term behaved as a regularization term, that force the sampled points to be not too far away from current area. This is because only when the selected points are closer to the current optimal point, are they more likely to learn which nearby area may

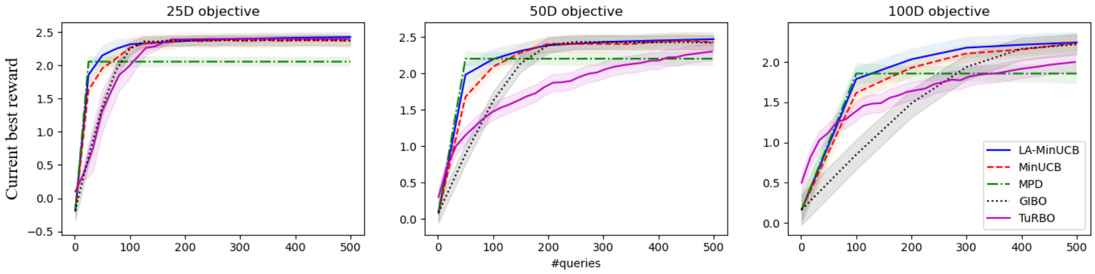

Figure 2: Progressive optimized reward on high-dimensional synthetic functions. LA-MinUCB demonstrates fast and accurate convergence compared to other methods.

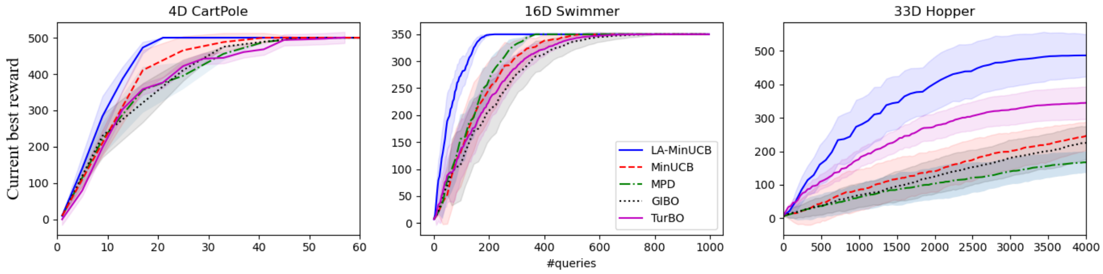

Figure 3: Progressive optimized reward on the MuJuCo tasks. LA-MinUCB has consistently optimal performance.

contain smaller values, thereby achieving a greater decrease in the next exploitation step. This extra standard derivation term restrict the over-exploration of the original Knowledge Gradient to help it perform better exploitation. Benefiting from this property, LA-MinUCB is able to focus on doing local search and quickly find local optima points.

The significant advantage of using LA-MinUCB is that its strategy has good theoretical properties:

Theorem 3. *If only one iteration is left and we can observe the function value through sampling, then the local exploration in LA-MinUCB is Bayes-optimal among all feasible policies.*

The proof of this theorem is listed in Appendix B.5. This indicates that LA-MinUCB is a greedy strategy, where in each step it use the optimal acquisition function to enhance the exploitation. From this perspective, LA-MinUCB is likely to be superior to MinUCB because LA-MinUCB is better prepared for the next step of exploitation. This is also reflected in our experiments results, which shows that LA-MinUCB have a very competitive performance in numerical experiments and practical applications. Meanwhile, this also illustrates that good local Bayesian optimization algorithms can be constructed without the need for approximate gradients.

8 Experiments

In this section we apply varies experimental settings to assess the efficacy of our proposed two algorithms, Min-UCB and LA-MinUCB, relative to two established methodologies. These include two approximated gradient methods, (1) GIBO [22] and (2)MPD [23], which has shown strong performance in local Bayesian optimization methods, and one trust region based method (3) TurBO [7]. Our code is based on Nguyen et al. [23], where they provide the program of GIBO, MPD, and various of objective functions. Our experimental settings on synthetic (Sect. 4.1) and reinforcement learning (Sect. 4.2) objectives are same as their papers. Each algorithm is executed a total of ten times for every objective function that we examine, initiating from an identical set of starting points sampled across the bounded domain via a Sobol sequence. We illustrate the results in Fig. (2) (synthetic objective) and (3) (reinforcement learning objective). The figures shows the mean of current best reward on the number of queries with an error bar (defined through the standard derivation). Our experimental framework was executed on a workstation of 20 Intel Xeon CPU cores, with a 32GB of memory. The codes can be viewed on https://github.com/chinafzy1/Minimizing-UCB. For more computational details, please refer to Appendix A. For more experimental results and ablation study, please refer to Appendix C.

8.1 Synthetic Objectives

In our first experiments, we focused on optimizing synthetic objective functions within the d-dimensional unit hypercube $[0, 1]^d$. These functions were generated by sampling from a Gaussian Process (GP) with a Radial Basis Function (RBF) kernel. This Synthetic objective is first mentioned in Müller et al. [22] , and for a more comprehensive understanding of this experimental setup, please refer to §4.1 in their work. Each experiment was allocated a budget of 500 function evaluations, under the dimension $\{25, 50, 100\}$ separately. The error bar here is defined as plus or minus 0.2 standard derivation for better illustration, as the functions sampled by Gaussian processes have significant differences. Fig. (2) illustrates that our proposed LA-MinUCB and Min-UCB achieve highly competitive performance in both efficiency and accuracy, where LA-MinUCB is slightly better. The MPD ascent very rapidly at the beginning, but soon shows instability. Although each ascent applies the direction of maximum probability, it considers multiple ascent in one iteration in the algorithm. This strategy may instead lead to a small probability that the final point is greater than the original point, causing the algorithm to fall into suboptimal solutions. GIBO can achieve very close to the real local optimum, but its strategy is too conservative, resulting in slow convergence. Similarly, when the dimensionality is particularly high, TurBO performs relatively poorly and does not have much competitiveness.

8.2 Reinforcement Learning Objective

In this experiment, we turned to reinforcement learning, specifically MuJoCo-based locomotion tasks [30]. In these tasks, we try to learn a linear policy that maps states into actions in order to maximize the rewards given by the learning environment. We adopt the experiments similar with Nguyen et al. [23] including three environments: CartPole-v1 with 4 parameters, Swimmer-v1 with 16 parameters, and Hopper-v1 with 33 parameters. The only difference is that we apply the Hopper-v1 without the state normalization, as in Nguyen's work they show that this state normalization will cause a significant unstable change in the function value. The error bar here is defined as plus or minus one standard derivation. Our results, depicted in Fig. (3), show that LA-MinUCB has an excellent performance in all three cases. LA-MinUCB exhibits faster convergence speed and may achieve a better local optima than other methods. Although MinUCB has also shown slightly better performance than GIBO, it is still not that efficient as LA-MinUCB. This reflects the powerful performance of LA-MinUCB in real applications, and also indicates that efficient local Bayesian optimization algorithm can be designed without the estimation of gradient information.

9 Conclusion

In this paper we find the connection between gradient descent and minimizing the UCB, demonstrating that minimizing UCB is an efficient local exploitation strategy with the Gaussian Processes surrogate. We introduce MinUCB, an algorithm applying minimizing the UCB as the objective of local exploitation, that will converges to local optima within a polynomial rate. By enhancing the local exploration acquisition function of MinUCB, we have developed a more advanced local Bayesian optimization algorithm, LA-MinUCB. We have tested our algorithm on various synthetic and reinforcement learning objectives, and the results have confirmed the efficacy of our approaches.

In this article, we have not provided the convergence proof of the LA-MinUCB algorithm. One consideration is whether LA-MinUCB have a better theoretical convergence rate compared to other methods, such as GIBO? Additionally, are there any other better local exploitation strategy in local Bayesian optimization? These will become interesting research directions in the future.

Acknowledgments and Disclosure of Funding

Zheyi Fan and Qingpei Hu's work are partly supported by National Key Research and Development Program of China (2021YFA1000300 and 2021YFA1000301)

Szu Hui Ng's work is supported in part by the Ministry of Education, Singapore (grant: R-266-000-149-114).

References

[1] Riad Akrour, Dmitry Sorokin, Jan Peters, and Gerhard Neumann. Local bayesian optimization of motor skills. In *International Conference on Machine Learning*, pages 41–50. PMLR, 2017.

[2] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

[3] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[4] Mickael Binois and Nathan Wycoff. A survey on high-dimensional gaussian process modeling with application to bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2(2):1–26, 2022.

[5] Josip Djolonga, Andreas Krause, and Volkan Cevher. High-dimensional gaussian process bandits. *Advances in neural information processing systems*, 26, 2013.

[6] David Eriksson and Martin Jankowiak. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

[7] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

[8] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[9] Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. Discovering and exploiting additive structure for bayesian optimization. In *Artificial Intelligence and Statistics*, pages 1311–1319. PMLR, 2017.

[10] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945, 2014.

[11] Michael A Gelbart, Jasper Snoek, and Ryan P Adams. Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 250–259, 2014.

[12] Jeff Guo, Bojana Ranković, and Philippe Schwaller. Bayesian optimization for chemical reactions. *Chimia*, 77(1/2):31–38, 2023.

[13] Eric Han, Ishank Arora, and Jonathan Scarlett. High-dimensional bayesian optimization via tree-structured additive models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7630–7638, 2021.

[14] Johannes Kirschner, Mojmir Mutny, Nicole Hiller, Rasmus Ischebeck, and Andreas Krause. Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. In *International Conference on Machine Learning*, pages 3429–3438. PMLR, 2019.

[15] Johannes Kirschner, Mojmir Mutnỳ, Andreas Krause, Jaime Coello de Portugal, Nicole Hiller, and Jochem Snuverink. Tuning particle accelerators with safety constraints using bayesian optimization. *Physical Review Accelerators and Beams*, 25(6):062802, 2022.

[16] Rémi Lam, Matthias Poloczek, Peter Frazier, and Karen E Willcox. Advances in bayesian optimization with applications in aerospace engineering. In *2018 AIAA Non-Deterministic Approaches Conference*, page 1656, 2018.

[17] Armin Lederer, Jonas Umlauft, and Sandra Hirche. Uniform error bounds for gaussian process regression with application to safe control. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional bayesian optimization. *Advances in neural information processing systems*, 33:1546–1558, 2020.

[19] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

[20] Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. *Advances in neural information processing systems*, 35:34505–34518, 2022.

[21] Riccardo Moriconi, Marc Peter Deisenroth, and KS Sesh Kumar. High-dimensional bayesian optimization using low-dimensional feature spaces. *Machine Learning*, 109:1925–1943, 2020.

[22] Sarah Müller, Alexander von Rohr, and Sebastian Trimpe. Local policy search with bayesian optimization. *Advances in Neural Information Processing Systems*, 34:20708–20720, 2021.

[23] Quan Nguyen, Kaiwen Wu, Jacob Gardner, and Roman Garnett. Local bayesian optimization via maximizing probability of descent. *Advances in neural information processing systems*, 35:13190–13202, 2022.

[24] Beth A Reid, Will J Percival, Daniel J Eisenstein, Licia Verde, David N Spergel, Ramin A Skibba, Neta A Bahcall, Tamas Budavari, Joshua A Frieman, Masataka Fukugita, et al. Cosmological constraints from the clustering of the sloan digital sky survey dr7 luminous red galaxies. *Monthly Notices of the Royal Astronomical Society*, 404(1):60–85, 2010.

[25] Shubhanshu Shekhar and Tara Javidi. Significance of gradient information in bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2836–2844. PMLR, 2021.

[26] Yu Shen, Yang Li, Jian Zheng, Wentao Zhang, Peng Yao, Jixiang Li, Sen Yang, Ji Liu, and Bin Cui. Proxybo: Accelerating neural architecture search via bayesian optimization with zero-cost proxies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9792–9801, 2023.

[27] Wei Shyy. A study of finite difference approximations to steady-state, convection-dominated flow problems. *Journal of Computational Physics*, 57(3):415–438, 1985.

[28] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[29] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022. Omnipress, 2010.

[30] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[31] Matteo Turchetta, Andreas Krause, and Sebastian Trimpe. Robust model-free reinforcement learning with multi-objective bayesian optimization. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 10702–10708. IEEE, 2020.

[32] Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In *International Conference on Machine Learning*, pages 10663–10674. PMLR, 2021.

[33] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018.

[34] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[35] Jian Wu, Matthias Poloczek, Andrew G Wilson, and Peter Frazier. Bayesian optimization with gradients. *Advances in neural information processing systems*, 30, 2017.

[36] Kaiwen Wu, Kyurae Kim, Roman Garnett, and Jacob Gardner. The behavior and convergence of local bayesian optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

[37] Juliusz Krzysztof Ziomek and Haitham Bou Ammar. Are random decompositions all we need in high dimensional bayesian optimisation? In *International Conference on Machine Learning*, pages 43347–43368. PMLR, 2023.

Algorithm 3: GIBO

```
Input: A black-box function $f$
for $t = 1, 2, ..., T$ do
    $\mathbf{X} = \arg\min_{\mathbf{Z}} \alpha_{trace}(\mathbf{x}_t, \mathbf{Z})$ where $\mathbf{Z} \in \mathbb{R}^{b_t \times d}$        #Local exploration (sampling)
    evaluate the black-box function $f$ on $\mathbf{X}$, obtaining noisy measurements $\mathbf{y}$
    $\mathcal{D}_t = \mathcal{D}_{t-1} \cup (\mathbf{X}, \mathbf{y})$
    $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \nabla \mu_{\mathcal{D}_t}(\mathbf{x}_t)$        #Local exploitation (step move)
end
```

A Additional details of the algorithms

In this section we will provide additional details about our proposed algorithms. First we list the GIBO algorithm in the version of Wu et al.[36], as shown in Algorithm 3. MinUCB adopt the same local exploration acquisition function to sample points as in GIBO, and change the objective of local exploitation acquisition function into minimizing the UCB, instead of gradient descent step.

As for the computational problem of LA-MinUCB, the optimization for the local exploration acquisition function is quite hard, as it involves a nested optimization problem. To handle this task we use the idea of one-shot optimization in BoTorch [2], that transform the original problem into a deterministic optimization problem with fixed sampling. This method is first proposed to solve the optimization of Knowledge Gradient. However, considering that our method is very similar to Knowledge Gradient, their optimization method is also applicable to our problem. For more information about this one-shot optimization method, please refer to the Section 4.2 in Balandat et al. [2].

In our numerical experiment, which has been shown in Section 8, we apply the same experimental settings with Müller et al. [22] and Nguyen et al. [23]. The coefficients in GIBO and MPD are chosen as the optimal one, which are reported in their experiments or code. As for our MinUCB and LA-MinUCB, we use a fixed batch size in each experiment, i.e. $b_t^{(1)}$, $b_t^{(2)}$ and b_t are unchanged in each experiment. The UCB coeffieient β^t in MinUCB and LA-MinUCB is fixed as 3, which has been shown to be able to balance the convergence speed and accuracy of the algorithm.

B Theoretical analysis for MinUCB and LA-MinUCB

In this section, we will present the theoretical results of the algorithms introduced in the article. Sections B.1-B.4 provide the theoretical convergence proof for MinUCB, while B.5 discusses the theoretical properties of LA-MinUCB. Sections B.1-B.3 serve as the preliminary theory for proving the convergence of MinUCB, mainly presenting the local smoothness properties of Gaussian processes, mean functions, and standard deviation functions, as well as the theoretical properties of approximate gradients. B.4 combines the above theories, connecting the minimization of UCB with gradient descent, and provides the convergence theorem for MinUCB. B.5 demonstrates that LA-MinUCB has one-step Bayesian optimality, illustrating the superiority of the algorithm.

B.1 The function smoothness in Gaussian Process and mean function

In this subsection we mainly build the theory for the smoothness of Gaussian process and mean function. Our convergence proof heavily depend on the smoothness analysis on the Gaussian process, mean function and standard derivation function. These proof rely on the upper bound of Gaussian processes proposed by Lederer et al.[17], which is related to the Lipshitz continues coefficient on kernel function:

Lemma 1 (Lederer et al.[17]). *Consider a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot, \cdot)$, and let L_k denote its Lipschitz constant on the set $\mathbb{X}$ with maximum extension $r = \max_{\mathbf{x},\mathbf{x}' \in \mathbb{X}} \|\mathbf{x} - \mathbf{x}^{'}\|_2$. Then, with probability of at least $1 - \delta$, the supremum of a sample function $f(\mathbf{x})$ of this Gaussian process is bounded by:*

$$\sup_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x}) \leq \sqrt{2\log\left(\frac{1}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k(\mathbf{x},\mathbf{x})} + 12\sqrt{6d} \max\left\{\max_{\mathbf{x}\in\mathcal{X}} \sqrt{k(\mathbf{x},\mathbf{x})}, \sqrt{rL_k}\right\}$$

We use this lemma to build the upper bound on the L-smoothness coefficient on the Gaussian process and mean function:

Theorem 3 (Smoothness of Gaussian process)**.** *Consider a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot,\cdot)$, and $k(\cdot,\cdot)$ satisfies Assumption 1. We define the second-order partial derivative $k^{\partial_{ij}}(\cdot,\cdot)$ as*

$$k^{\partial_{ij}}(\mathbf{x},\mathbf{x}^{'}) = \frac{\partial^4}{\partial x_i \partial x_j \partial x_i^{'} \partial x_j^{'}} k(\mathbf{x},\mathbf{x}^{'})$$

Let $L_k^{\partial_{ij}}$ denote the Lipschitz constants of the second-order partial derivative $k^{\partial_{ij}}(\cdot,\cdot)$ on the set $\mathbb{X}$ with maximum extension $r = \max_{\mathbf{x},\mathbf{x}'\in\mathbb{X}} \|\mathbf{x}-\mathbf{x}^{'}\|_2$. Then, a sample function $f(\cdot)$ is β-smooth with probability of at least $1-\delta$, and β satisfies:

$$\beta \leq \sqrt{\sum_{i,j} U_{ij}^2}$$

where

$$U_{ij} = \sqrt{2\log\left(\frac{2d^2}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + 12\sqrt{6d}\max\left\{\max_{\mathbf{x}\in\mathbb{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})}, \sqrt{rL_k^{\partial_{ij}}}\right\}$$

Proof. We set the Hessian matrix of $f(\mathbf{x})$ as

$$H(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} f(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_d} f(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_d \partial x_1} f(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_d \partial x_d} f(\mathbf{x}) \end{bmatrix}$$

According to the equivalence of matrix norms, we have

$$\|H(\mathbf{x})\|_2 \leq \|H(\mathbf{x})\|_F$$

Thus

$$\beta \leq \sup_{\mathbf{x}\in\mathbb{X}} \|H(\mathbf{x})\|_2 \leq \sup_{\mathbf{x}\in\mathbb{X}} \|H(\mathbf{x})\|_F = \sup_{\mathbf{x}\in\mathbb{X}} \sqrt{\sum_{i,j}^{d} (\frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x}))^2} \tag{8}$$

Then through applying Lemma 2, we have the following equation holds with probability at least $1-\delta/d^2$

$$\sup_{\mathbf{x}\in\mathbb{X}} \left| \frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x}) \right| \leq \sqrt{2\log\left(\frac{2d^2}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + 12\sqrt{6d}\max\left\{\max_{\mathbf{x}\in\mathbb{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})}, \sqrt{rL_k^{\partial_{ij}}}\right\} \tag{9}$$

Then if we combine the Eq.(8) and Eq.(9), we will get the result of Theorem 1. □

Theorem 4. *If $\sup_{\mathbf{x}\in\mathbb{X}} |f(\mathbf{x})| < L$, then with the probability of at least $1-\delta$, for a dataset $\mathcal{D}$, we have*

$$\max_{\mathbf{x}\in\mathbb{X}} |\mu_{\mathcal{D}}(\mathbf{x})| \leq \sqrt{2\log\left(\frac{1}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k(\mathbf{x},\mathbf{x})} + L$$

Proof. For a dataset $\mathcal{D}$, we have the mean function $\mu_{\mathcal{D}}(\mathbf{x})$ and variance function $\sigma_{\mathcal{D}}^2(\mathbf{x})$.

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbb{X}} |\mu_{\mathcal{D}}(\mathbf{x})|$$

$$\begin{aligned} |\mu_{\mathcal{D}}(\mathbf{x}^*)| &\leq |f(\mathbf{x}^*)| + |\mu_{\mathcal{D}}(\mathbf{x}^*) - f(\mathbf{x}^*)| \\ &\leq \sup_{\mathbf{x}\in\mathbb{X}} |f(\mathbf{x})| + |\mu_{\mathcal{D}}(\mathbf{x}^*) - f(\mathbf{x}^*)| \end{aligned} \tag{10}$$

For a random variable $r \sim \mathcal{N}(0,1)$, we have $P(|r| > c) \leq \exp(-\frac{c^2}{2})$. Thus with the probability of at least $1 - \frac{\delta}{2}$, we have

$$|\mu_{\mathcal{D}}(\mathbf{x}^*) - f(\mathbf{x}^*)| \leq \sqrt{2\log\frac{1}{\delta}}\sigma_{\mathcal{D}}(\mathbf{x}^*) \leq \sqrt{2\log\frac{1}{\delta}}\max_{\mathbf{x}\in\mathbb{X}}\sqrt{k(\mathbf{x},\mathbf{x})} \tag{11}$$

Combine the result of Eq(10) and Eq(11), we have the following results with the probability of at least $1-\delta$:

$$\max_{\mathbf{x}\in\mathbb{X}}|\mu_{\mathcal{D}}(\mathbf{x})| \leq \sqrt{2\log\left(\frac{1}{\delta}\right)}\max_{\mathbf{x}\in\mathbb{X}}\sqrt{k(\mathbf{x},\mathbf{x})} + L$$

□

Theorem 5 (Smoothness of mean function)**.** *Consider a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot,\cdot)$, and $k(\cdot,\cdot)$ satisfies Assumption 1. Let $L_k^{\partial_{ij}}$ denote the Lipschitz constants of the second-order partial derivative $k^{\partial_{ij}}(\cdot,\cdot)$ on the set $\mathbb{X}$ with maximum extension $r = \max_{\mathbf{x},\mathbf{x}'\in\mathbb{X}}\|\mathbf{x}-\mathbf{x}'\|_2$. Then, given a dataset $\mathcal{D}$, the mean function $\mu_{\mathcal{D}}(\mathbf{x})$ is β_μ-smooth with probability of at least $1-\delta$, and β satisfies:*

$$\beta_\mu \leq \sqrt{\sum_{i,j} V_{ij}^2}$$

where

$$V_{ij} = \sqrt{2\log\left(\frac{d^2}{\delta}\right)}\max_{\mathbf{x}\in\mathbb{X}}\sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + L_{ij}$$

and $L_{ij} = \sup_{\mathbf{x}\in\mathbb{X}}\left|\frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x})\right|$

Proof. We set the Hessian matrix of $\mu_{\mathcal{D}}(\mathbf{x})$ as

$$H_\mu(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial x_1}\mu_{\mathcal{D}}(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_d}\mu_{\mathcal{D}}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_d \partial x_1}\mu_{\mathcal{D}}(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_d \partial x_d}\mu_{\mathcal{D}}(\mathbf{x}) \end{bmatrix}$$

According to the equivalence of matrix norms, we have

$$\|H_\mu(\mathbf{x})\|_2 \leq \|H_\mu(\mathbf{x})\|_F$$

Thus we can apply Theorem 3 and get the following result probability of at least $1-\delta$:

$$\beta_\mu \leq \sup_{\mathbf{x}\in\mathbb{X}}\|H_\mu(\mathbf{x})\|_2 \leq \sup_{\mathbf{x}\in\mathbb{X}}\|H_\mu(\mathbf{x})\|_F = \max_{\mathbf{x}\in\mathbb{X}}\sqrt{\sum_{i,j}^{d}(\frac{\partial^2}{\partial x_i \partial x_j}\mu_{\mathcal{D}}(\mathbf{x}))^2} \leq \sqrt{\sum_{i,j}^{d}\max_{\mathbf{x}\in\mathbb{X}}(\frac{\partial^2}{\partial x_i \partial x_j}\mu_{\mathcal{D}}(\mathbf{x}))^2} \tag{12}$$

where

$$\max_{\mathbf{x}\in\mathbb{X}}\left|\frac{\partial^2}{\partial x_i \partial x_j}\mu_{\mathcal{D}}(\mathbf{x})\right| \leq \sqrt{2\log\left(\frac{d^2}{\delta}\right)}\max_{\mathbf{x}\in\mathbb{X}}\sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + L_{ij}$$

□

Here we need an additional Theorem to give the upper bound on the L-smoothness coefficient on the posterior of $f(\cdot)$, which is aim to bound the probability under a specific L-smoothness coefficient.

Lemma 2. *Consider a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot,\cdot)$, and let L_k denote its Lipschitz constant on the set $\mathbb{X}$ with maximum extension $r = \max_{\mathbf{x},\mathbf{x}'\in\mathbb{X}} \|\mathbf{x}-\mathbf{x}^{'}\|_2$. Then, given a dataset $\mathcal{D}$, we define the posterior of $f(\cdot)$ over dataset $\mathcal{D}$ as $f_{\mathcal{D}}(\cdot)$. With probability of at least $1-\delta$, the supremum of a sample function $f_{\mathcal{D}}(\mathbf{x})$ of this Gaussian process is bounded by:*

$$\sup_{\mathbf{x}\in\mathbb{X}} f_{\mathcal{D}}(\mathbf{x}) \leq \sqrt{2\log\left(\frac{1}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k(\mathbf{x},\mathbf{x})} + 12\sqrt{6d}\max\left\{\max_{\mathbf{x}\in\mathcal{X}} \sqrt{k(\mathbf{x},\mathbf{x})}, \sqrt{rL_k}\right\}$$

Proof. Note that for the covariance pseudo-metric $d_k(x,x') = \sqrt{k(x,x)+k(x',x')-2k(x,x')}$, we have

$$\begin{aligned}
k_{\mathcal{D}}(\mathbf{x},\mathbf{x}) + k_{\mathcal{D}}(\mathbf{x}',\mathbf{x}') - 2k_{\mathcal{D}}(\mathbf{x},\mathbf{x}') =& k(\mathbf{x},\mathbf{x}) + k(\mathbf{x}',\mathbf{x}') - 2k(\mathbf{x},\mathbf{x}') \\
&- (k(\mathbf{x},\mathbf{X}_{\mathcal{D}}) - k(\mathbf{x}^{'},\mathbf{X}_{\mathcal{D}}))(k(\mathbf{X}_{\mathcal{D}},\mathbf{X}_{\mathcal{D}}) + \sigma^2 I)^{-1}(k(\mathbf{x},\mathbf{X}_{\mathcal{D}}) - k(\mathbf{x}^{'},\mathbf{X}_{\mathcal{D}}))^T \\
\leq& k(\mathbf{x},\mathbf{x}) + k(\mathbf{x}',\mathbf{x}') - 2k(\mathbf{x},\mathbf{x}')
\end{aligned}$$

Then this Lemma can be proved in a same way with Lemma B.1 in Lederer et al. [17]. □

Theorem 6 (Smoothness of the posterior Gaussian process)**.** *Consider a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot,\cdot)$, and $k(\cdot,\cdot)$ satisfies Assumption 1. Let $L_k^{\partial_{ij}}$ denote the Lipschitz constants of the second-order partial derivative $k^{\partial_{ij}}(\cdot,\cdot)$ on the set $\mathbb{X}$ with maximum extension $r = \max_{\mathbf{x},\mathbf{x}'\in\mathbb{X}} \|\mathbf{x}-\mathbf{x}^{'}\|_2$. Then, a sample function $f_{\mathcal{D}}(\cdot)$ is $\beta_{\mathcal{D}}$-smooth with probability of at least $1-\delta$, and β satisfies:*

$$\beta_{\mathcal{D}} \leq \sqrt{\sum_{i,j} \tilde{U}_{ij}^2}$$

where

$$\tilde{U}_{ij} = L_{ij} + 2\sqrt{2\log\left(\frac{4d^2}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + 12\sqrt{6d}\max\left\{\max_{\mathbf{x}\in\mathcal{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})}, \sqrt{rL_k^{\partial_{ij}}}\right\}$$

and $L_{ij} = \sup_{\mathbf{x}\in\mathbb{X}} \left|\frac{\partial^2}{\partial x_i \partial x_j} f(\mathbf{x})\right|$

Proof. We set the Hessian matrix of $f_{\mathcal{D}}(\mathbf{x})$ as

$$H_{\mathcal{D}}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} f_{\mathcal{D}}(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_d} f_{\mathcal{D}}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_d \partial x_1} f_{\mathcal{D}}(\mathbf{x}) & \cdots & \frac{\partial^2}{\partial x_d \partial x_d} f_{\mathcal{D}}(\mathbf{x}) \end{bmatrix}$$

$$\beta_{\mathcal{D}} \leq \sup_{\mathbf{x}\in\mathbb{X}} \|H_{\mathcal{D}}(\mathbf{x})\|_2 \leq \sup_{\mathbf{x}\in\mathbb{X}} \|H_{\mathcal{D}}(\mathbf{x})\|_F = \sup_{\mathbf{x}\in\mathbb{X}} \sqrt{\sum_{i,j}^{d} (\frac{\partial^2}{\partial x_i \partial x_j} f_{\mathcal{D}}(\mathbf{x}))^2} \leq \sqrt{\sum_{i,j}^{d} \sup_{\mathbf{x}\in\mathbb{X}} (\frac{\partial^2}{\partial x_i \partial x_j} f_{\mathcal{D}}(\mathbf{x}))^2} \tag{13}$$

The posterior $f_{\mathcal{D}}(\cdot)$ can be divided into two parts:

$$\sup_{\mathbf{x}\in\mathbb{X}} \left|\frac{\partial^2}{\partial x_i \partial x_j} f_{\mathcal{D}}(\mathbf{x})\right| \leq \sup_{\mathbf{x}\in\mathbb{X}} \left|\frac{\partial^2}{\partial x_i \partial x_j} \mu_{\mathcal{D}}(\mathbf{x})\right| + \sup_{\mathbf{x}\in\mathbb{X}} \left|\frac{\partial^2}{\partial x_i \partial x_j} f_{\mathcal{D}}(\mathbf{x}) - \frac{\partial^2}{\partial x_i \partial x_j} \mu_{\mathcal{D}}(\mathbf{x})\right| \tag{14}$$

where $\frac{\partial^2}{\partial x_i \partial x_j} f_{\mathcal{D}}(\mathbf{x}) - \frac{\partial^2}{\partial x_i \partial x_j} \mu_{\mathcal{D}}(\mathbf{x})$ is a zero mean Gaussian process with covariance function $k_{\mathcal{D}}^{\partial ij}(\mathbf{x},\mathbf{x}^{'})$. According to Lemma 2, we have the following result with probability at least $1 - \frac{\delta}{2d^2}$

$$\begin{aligned}
&\sup_{\mathbf{x}\in\mathbb{X}} \left|\frac{\partial^2}{\partial x_i \partial x_j} f_{\mathcal{D}}(\mathbf{x}) - \frac{\partial^2}{\partial x_i \partial x_j} \mu_{\mathcal{D}}(\mathbf{x})\right| \\
&\quad \leq \sqrt{2\log\left(\frac{4d^2}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + 12\sqrt{6d}\max\left\{\max_{\mathbf{x}\in\mathcal{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})}, \sqrt{rL_k^{\partial_{ij}}}\right\}
\end{aligned} \tag{15}$$

Through the result of Lemma 4, with the probability of at least $1-\frac{\delta}{2d^2}$,

$$\max_{\mathbf{x}\in\mathbb{X}}\left|\frac{\partial^2}{\partial x_i\partial x_j}\mu_{\mathcal{D}}(\mathbf{x})\right| \leq \sqrt{2\log\left(\frac{2d^2}{\delta}\right)}\max_{\mathbf{x}\in\mathbb{X}}\sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})}+L_{ij} \tag{16}$$

Combine the result of Eq (13), Eq (14), Eq (15) and Eq (16), this Theorem is proved. □

B.2 Local smoothness in standard derivation function

In this subsection we attempt to build the smoothness theorem for standard derivation function $\sigma_{\mathcal{D}}(\mathbf{x})$. However, $\sigma_{\mathcal{D}}(\mathbf{x})$ is not L-smoothness under some common kernels such as Gaussian kernel or Matérn kernel with $\gamma = 2.5$. However, we can proof that $\sigma_{\mathcal{D}}(\mathbf{x})$ may achieve a similar result with some small error term.

Here we define an event $U_L = \{f(\cdot)|f(\cdot)\text{ is L-smooth}\}$, and $\sigma_{\mathcal{D}}(\mathbf{x}|L) = \sqrt{\frac{\pi}{2}}E(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|\big|\mathcal{D}, f(\cdot)\in U_L)$

Lemma 3. *For any $\mathbf{x}\in\mathbb{X}$ and dataset $\mathcal{D}$, we have*

$$\sigma_{\mathcal{D}}(\mathbf{x}|L)\leq\sigma_{\mathcal{D}}(\mathbf{x})+(\frac{1}{P(U_L|\mathcal{D})}-1)\max_{\mathbf{x}\in\mathbb{R}^d}\sqrt{k(\mathbf{x},\mathbf{x})}$$

Proof. According to the definition of $\sigma_L(\mathbf{x})$, we have

$$P(U_L|\mathcal{D})\sigma_{\mathcal{D}}(\mathbf{x}|L)+P(U_L^c|\mathcal{D})\sqrt{\frac{\pi}{2}}E(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|\big|\mathcal{D},f(\cdot)\in U_L^c)=\sqrt{\frac{\pi}{2}}E(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|\big|\mathcal{D})=\sigma_{\mathcal{D}}(\mathbf{x})$$

Thus

$$\begin{aligned}\sigma_{\mathcal{D}}(\mathbf{x}|L)&\leq\frac{1}{P(U_L|\mathcal{D})}\sigma_{\mathcal{D}}(\mathbf{x})\\&=\sigma_{\mathcal{D}}(\mathbf{x})+(\frac{1}{P(U_L|\mathcal{D})}-1)\sigma_{\mathcal{D}}(\mathbf{x})\\&\leq\sigma_{\mathcal{D}}(\mathbf{x})+(\frac{1}{P(U_L|\mathcal{D})}-1)\max_{\mathbf{x}\in\mathbb{R}^d}\sqrt{k(\mathbf{x},\mathbf{x})}\end{aligned}$$

□

Lemma 4. *If $P(U_L|\mathcal{D})>\frac{1}{2}$, there exist constants $c_1>1$ and $c_2>0$ independent of $\mathbf{x}$ and dataset $\mathcal{D}$ that satisfies*

$$\sigma_{\mathcal{D}}(\mathbf{x})\leq c_1\sigma_{\mathcal{D}}(\mathbf{x}|L)+c_2P(U_L^c|\mathcal{D})$$

Proof. According to the Markov inequality, we have

$$P(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|>a\big|\mathcal{D},f(\cdot)\in U_L)\leq\frac{E(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|\big|\mathcal{D},f(\cdot)\in U_L)}{a}=\sqrt{\frac{2}{\pi}}\frac{\sigma_{\mathcal{D}}(\mathbf{x}|L)}{a}$$

For any event A and B, we have $P(A\cap B)=P(A)-P(A\cap B^c)\geq P(A)-P(B^c)$. Thus we have

$$\begin{aligned}P(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|>a\big|\mathcal{D},f(\cdot)\in U_L)&=\frac{P(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|>a,f(\cdot)\in U_L\big|\mathcal{D})}{P(U_L|\mathcal{D})}\\&\geq\frac{P(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|>a\big|\mathcal{D})-P(U_L^c|\mathcal{D})}{P(U_L|\mathcal{D})}\end{aligned}$$

If we set $a=\sigma_{\mathcal{D}}(\mathbf{x})$, we will have the following inequality

$$\begin{aligned}\sigma_{\mathcal{D}}(\mathbf{x}|L)&\geq a\sqrt{\frac{\pi}{2}}P(|f(\mathbf{x})-\mu_{\mathcal{D}}(\mathbf{x})|>a\big|\mathcal{D},f(\cdot)\in U_L)\\&\geq\sqrt{\frac{\pi}{2}}\frac{2-2\Phi(1)-P(f(\cdot)\in U_L^c|\mathcal{D})}{P(U_L|\mathcal{D})}\sigma_{\mathcal{D}}(\mathbf{x})\\&\geq(2-2\Phi(1))\sqrt{\frac{\pi}{2}}\sigma_{\mathcal{D}}(\mathbf{x})-2\sqrt{\frac{\pi}{2}}\max_{\mathbf{x}\in\mathbb{X}}\sqrt{k(\mathbf{x},\mathbf{x})}P(U_L^c|\mathcal{D})\end{aligned}$$

So through the above analysis, we have

$$\sigma_{\mathcal{D}}(\mathbf{x}) \leq c_1 \sigma_{\mathcal{D}}(\mathbf{x}|L) + c_2 P(U_L^c|\mathcal{D})$$

where $c_1 = \sqrt{\frac{2}{\pi}} \frac{1}{2-2\Phi(1)}$, $c_2 = \frac{1}{1-\Phi(1)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k(\mathbf{x},\mathbf{x})}$ □

Lemma 5. *If $\mu_{\mathcal{D}}(\cdot)$ is L_μ smooth and $P(U_L|\mathcal{D}) > \frac{1}{2}$, then we have*

$$|\sigma_{\mathcal{D}}(\mathbf{x}_1|L) - \sigma_{\mathcal{D}}(\mathbf{x}_2|L)| \leq \sqrt{\frac{\pi}{2}} \left(\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \sqrt{\boldsymbol{tr}(\nabla k_{\mathcal{D}}(\mathbf{x}_2, \mathbf{x}_2)\nabla^T)} + \frac{3L + 3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \right)$$

Proof.

$$\begin{aligned}
&\sqrt{\frac{2}{\pi}} |\sigma_{\mathcal{D}}(\mathbf{x}_1|L) - \sigma_{\mathcal{D}}(\mathbf{x}_2|L)| \\
&\qquad = |E(|f(\mathbf{x}_1) - \mu_{\mathcal{D}}(\mathbf{x}_1)\big|\mathcal{D}, f(\cdot) \in U_L) - E(|f(\mathbf{x}_2) - \mu_{\mathcal{D}}(\mathbf{x}_2)|\big|\mathcal{D}, f(\cdot) \in U_L)| \\
&\qquad \leq E(|f(\mathbf{x}_1) - \mu_{\mathcal{D}}(\mathbf{x}_1) - f(\mathbf{x}_2) + \mu_{\mathcal{D}}(\mathbf{x}_2)|\big|\mathcal{D}, f(\cdot) \in U_L)
\end{aligned}$$

As $f(\cdot) \in U_L$ and $\mu_{\mathcal{D}}(\cdot)$ is L_μ smooth,

$$\begin{aligned}
f(\mathbf{x}_1) &\leq f(\mathbf{x}_2) + < \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 > + \frac{L}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\
\mu_{\mathcal{D}}(\mathbf{x}_2) &\leq \mu_{\mathcal{D}}(\mathbf{x}_1) + < \nabla \mu_{\mathcal{D}}(\mathbf{x}_1), \mathbf{x}_2 - \mathbf{x}_1 > + \frac{L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\
&\leq \mu_{\mathcal{D}}(\mathbf{x}_1) + < \nabla \mu_{\mathcal{D}}(\mathbf{x}_2), \mathbf{x}_2 - \mathbf{x}_1 > + \frac{3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2
\end{aligned}$$

$$\begin{aligned}
f(\mathbf{x}_1) - \mu_{\mathcal{D}}(\mathbf{x}_1) - f(\mathbf{x}_2) + \mu_{\mathcal{D}}(\mathbf{x}_2) &\leq < \nabla f(\mathbf{x}_2) - \nabla \mu_{\mathcal{D}}(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 > + \frac{L + 3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\
&\leq \|\nabla f(\mathbf{x}_2) - \nabla \mu_{\mathcal{D}}(\mathbf{x}_2)\|_2 \|\mathbf{x}_1 - \mathbf{x}_2\|_2 + \frac{L + 3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2
\end{aligned}$$

In a similar way, we can prove

$$f(\mathbf{x}_1) - \mu_{\mathcal{D}}(\mathbf{x}_1) - f(\mathbf{x}_2) + \mu_{\mathcal{D}}(\mathbf{x}_2) \geq -\|\nabla f(\mathbf{x}_2) - \nabla \mu_{\mathcal{D}}(\mathbf{x}_2)\|_2 \|\mathbf{x}_1 - \mathbf{x}_2\|_2 - \frac{3L + L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2$$

So through the above analysis, we have the following results

$$\begin{aligned}
&\sqrt{\frac{2}{\pi}} \sigma_{\mathcal{D}}(\mathbf{x}_1|L) - \sigma_{\mathcal{D}}(\mathbf{x}_2|L)| \\
&\qquad \leq E(|f(\mathbf{x}_1) - \mu_{\mathcal{D}}(\mathbf{x}_1) - f(\mathbf{x}_2) + \mu_{\mathcal{D}}(\mathbf{x}_2)|\big|\mathcal{D}, f(\cdot) \in U_L) \\
&\qquad \leq E(\|\nabla f(\mathbf{x}_2) - \nabla \mu_{\mathcal{D}}(\mathbf{x}_2)\|_2|\big|\mathcal{D}, f(\cdot) \in U_L) \|\mathbf{x}_1 - \mathbf{x}_2\|_2 + \frac{3L + 3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\
&\qquad \leq \frac{E(\|\nabla f(\mathbf{x}_2) - \nabla \mu_{\mathcal{D}}(\mathbf{x}_2)\|_2 \big|\mathcal{D})}{P(U_L|\mathcal{D})} \|\mathbf{x}_1 - \mathbf{x}_2\|_2 + \frac{3L + 3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\
&\qquad \leq \frac{\sqrt{E(\|\nabla f(\mathbf{x}_2) - \nabla \mu_{\mathcal{D}}(\mathbf{x}_2)\|_2^2 |\mathcal{D})}}{P(U_L|\mathcal{D})} \|\mathbf{x}_1 - \mathbf{x}_2\|_2 + \frac{3L + 3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\
&\qquad \leq 2\sqrt{E(\|\nabla f(\mathbf{x}_2) - \nabla \mu_{\mathcal{D}}(\mathbf{x}_2)\|_2^2 |\mathcal{D})} \|\mathbf{x}_1 - \mathbf{x}_2\|_2 + \frac{3L + 3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 \\
&\qquad = \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \sqrt{\mathbf{tr}(\nabla k_{\mathcal{D}}(\mathbf{x}_2, \mathbf{x}_2)\nabla^T)} + \frac{3L + 3L_\mu}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2
\end{aligned}$$

The second to the last line is because of Cauchy-Schwarz inequality: $E(|X|) = E(|X| \cdot 1) \leq \sqrt{E(X^2)} \cdot 1 = \sqrt{E(X^2)}$. □

B.3 Some properties about approximate gradient descent

In this subsection we list properties about approximate gradient descent, which is mainly from Wu et al.[36]. This part is also essential in our final convergence proof, as we will need to connect the previous approximate gradient descent method with our minimizing UCB to give the convergence speed of gradient.

We first borrow the definition of Error function from Wu's work. The Error function measures the maximum reduction of uncertainty about the gradient estimation at $\mathbf{x} = 0$ when there are b data points $\mathbf{Z}$ without any extra dataset:

Definition 3. *(Error function) Given input dimensionality d, kernel k and noise standard deviation σ, we define the following error function:*

$$E_{d,k,\sigma}(b) = \inf_{\mathcal{Z}\in\mathbb{R}^{b\times d}} \boldsymbol{tr}(\nabla k(\mathbf{0},\mathbf{0})\nabla^T - \nabla k(\mathbf{0},\mathbf{Z})(k(\mathbf{Z},\mathbf{Z}) + \sigma^2 I)^{-1} k(\mathbf{0},\mathbf{Z})\nabla^T) \tag{17}$$

This Error function actually bounds the variance of the estimated gradient, which can be seen in the following lemma:

Lemma 6. *In the t^{th} iterations in MinUCB, we have*

$$\boldsymbol{tr}(\nabla k_{\mathcal{D}_t}(\mathbf{x}_t,\mathbf{x}_t)\nabla^T) \le E_{d,k,\sigma}(b_t^{(2)})$$

Proof. In the t^{th} step of MinUCB, the sampled candidates can be divided into two parts. One of them are sampled through local exploration acquisition function:

$$\mathbf{X} = \arg\min_{\mathbf{Z}} \alpha_{trace}(\mathbf{x}_t, \mathbf{Z})$$

Suppose their corresponding label is $\mathbf{y}$, and we set $\mathcal{D}_{2t} = (\mathbf{X}, \mathbf{y})$, then we can obtain:

$$\mathbf{tr}(\nabla k_{\mathcal{D}_t}(\mathbf{x}_t,\mathbf{x}_t)\nabla^T) \le \mathbf{tr}(\nabla k_{\mathcal{D}_{2t}}(\mathbf{x}_t,\mathbf{x}_t)\nabla^T)$$

The above inequality is mainly because the $\mathcal{D}_{2t}$ is the subset of $\mathcal{D}_t$. Then through the same analysis with Lemma 8 in Wu et al.[36], we can prove that

$$\mathbf{tr}(\nabla k_{\mathcal{D}_{2t}}(\mathbf{x}_t,\mathbf{x}_t)\nabla^T) \le E_{d,k,\sigma}(b_t^{(2)})$$

which complete our proof. □

To finish the convergence proof of MinUCB, we need the following lemmas from Wu et al.[36], this lemmas give the upper bound of the estimation error of gradient, and the upper bound of Error function under two common kernels RBF kernel and Matérn kernel with $\gamma = 2.5$.

Lemma 7. *For any $0 \le \delta \le 1$, let $C_t = 2\log\left(\frac{\pi^2 t^2}{6\delta}\right)$. Then the inequalities*

$$\|\nabla f(\mathbf{x}_t) - \nabla \mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 \le C_t \boldsymbol{tr}(\nabla k_{\mathcal{D}_t}(\mathbf{x}_t,\mathbf{x}_t)\nabla^T) \tag{18}$$

Lemma 8. *let $k(\mathbf{x}_1,\mathbf{x}_2) = \exp(-\frac{\|\mathbf{x}_1,\mathbf{x}_2\|_2}{2})$ be the RBF kernel. We have*

$$E_{d,k,\sigma}(2md) = O(\sigma d m^{-\frac{1}{2}})$$

Lemma 9. *let $k(\cdot,\cdot)$ be the Matérn kernel. Then we have*

$$E_{d,k,\sigma}(2md) = O(\sigma d m^{-\frac{1}{2}})$$

B.4 Convergence proof of MinUCB

In this part we will show the convergence of MinUCB, which will use all the results in previous Subsections. The difficulty in proving the convergence is to build the relationship with previous approximate gradient methods and minimizing UCB. In this proof we try to connect the function value on the gradient descent point and minimizing UCB point, and will need some smoothness properties on the Gaussian process, mean function and standard derivation function, which are provided in Subsection B.1 and B.2. To give the accurate convergence rate, we need the upper bound on the Error function, which is provided in Subsection B.3. We first give the basic convergence theorem on the gradient for MinUCB:

Theorem 1. *Suppose f is sampled from a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot,\cdot)$, and $k(\cdot,\cdot)$ satisfies Assumption 1. Then after t iterations of MinUCB algorithm, with the batch size $b_t^{(1)}$ and $b_t^{(2)}$, it satisfies that*

$$\min_{T/2\le t\le T} \|\nabla f(\mathbf{x}_t)\|_2 \le \frac{1}{\sqrt{\eta_T}}\sqrt{\frac{8}{T}\sum_{t=1}^{T}\frac{\tilde{\beta}_t\sigma}{\sqrt{b_t^{(1)}}} + \frac{8\pi}{T}\sum_{t=1}^{T}\tilde{\beta}_t^2\eta_t E_{d,k,\sigma}(b_t^{(2)}) + O(\frac{1}{T}\log\frac{1}{\delta})}$$
$$+\frac{1}{\sqrt{\eta_T}}\tilde{\beta}_{T/2}\sqrt{\frac{\pi}{2}}\sqrt{E_{d,k,\sigma}(b_{T/2}^{(2)})}$$

where $\tilde{\beta}_t$ and η_t are both decreasing sequence. They satisfies $\tilde{\beta}_t = O(\beta_t)$ and $\frac{1}{\eta_t} = O(d\sqrt{\log\frac{t^2d^2}{\delta}} + d^{\frac{3}{2}})$, and $\beta_t = \sqrt{2\log\frac{\pi^2t^2}{\delta}}$

Proof. According to the definition of $\mathbf{x}_{t+1}$, $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathbb{X}} \mu_{\mathcal{D}_t}(\mathbf{x}) + \beta_t\sigma_{\mathcal{D}_t}(\mathbf{x})$.

$$\begin{aligned} f(\mathbf{x}_{t+1}) &\le \min_{x\in\mathbb{R}^d} \mu_{\mathcal{D}_t}(\mathbf{x}) + \beta_t\sigma_{\mathcal{D}_t}(\mathbf{x}) \\ &\le \mu_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}) + \beta_t\sigma_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}) \end{aligned} \tag{19}$$

Where $\hat{\mathbf{x}}_{t+1}$ is a special point $\hat{\mathbf{x}}_{t+1} = \mathbf{x}_t - \eta_t\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)$, which is a pseudo gradient descent step. We will use this $\hat{\mathbf{x}}_{t+1}$ to build the connection between gradient descent and minimizing UCB. The β_t here is carefully chosen as $\beta_t = \sqrt{2\log\frac{\pi^2t^2}{\delta}}$ to guarantee

$$\sum_{t=1}^{\infty} P\big(f(\mathbf{x}_t) \le \mu_{\mathcal{D}_{t-1}}(\mathbf{x}_t) + \beta_t\sigma_{\mathcal{D}_{t-1}}(\mathbf{x}_t)\big) \ge 1 - \frac{\delta}{6}$$

Now we try to give the relationship between the UCB bound on $\mathbf{x}_t$ and $\hat{\mathbf{x}}_{t+1}$. Suppose the mean function $\mu_{\mathcal{D}_t}(\cdot)$ is $L_{\mu_t}-$smoothness (this coefficient will be given in the subsequent parts), we will have

$$\begin{aligned} \mu_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}) &\le \mu_{\mathcal{D}_t}(\mathbf{x}_t) + <\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t), \hat{\mathbf{x}}_{t+1} - \mathbf{x}_t> + \frac{L_{\mu_t}}{2}\|\hat{\mathbf{x}}_{t+1} - \mathbf{x}_t\|_2^2 \\ &\le \mu_{\mathcal{D}_t}(\mathbf{x}_t) - \eta_t\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 + \frac{L_{\mu_t}}{2}\eta_t^2\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 \end{aligned} \tag{20}$$

Here we apply the results in Subsection B.2, and give the upper bound for the $\sigma_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1})$. To simplify the symbols, we define $\mathbf{tr}(\nabla k_{\mathcal{D}_t}(\mathbf{x}_t,\mathbf{x}_t)\nabla^T) = \gamma_t$.

$$\begin{aligned} \sigma_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}) \le& c_1\sigma_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}|L_t) + c_2P(U_{L_t}^c|\mathcal{D}_t) \\ \le& c_1\sigma_{\mathcal{D}_t}(\mathbf{x}_t|L_t) + c_1\big|\sigma_{\mathcal{D}_t}(\hat{\mathbf{x}}_{t+1}|L_t) - \sigma_{\mathcal{D}_t}(\mathbf{x}_t|L_t)\big| + c_2P(U_{L_t}^c|\mathcal{D}_t) \\ \le& c_1\sigma_{\mathcal{D}_t}(\mathbf{x}_t|L_t) + c_1\sqrt{\frac{\pi}{2}}\eta_t\sqrt{\gamma_t}\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2 + c_1\sqrt{\frac{\pi}{2}}\frac{3L_t + 3L_{\mu_t}}{2}\eta_t^2\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 + c_2P(U_{L_t}^c|\mathcal{D}_t) \\ \le& c_1\sigma_{\mathcal{D}_t}(\mathbf{x}_t) + c_1\sqrt{\frac{\pi}{2}}\eta_t\sqrt{\gamma_t}\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2 + c_1\sqrt{\frac{\pi}{2}}\frac{3L_t + 3L_{\mu_t}}{2}\eta_t^2\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 \\ &+ c_2P(U_{L_t}^c|\mathcal{D}_t) + c_1(\frac{1}{P(U_{L_t}|\mathcal{D})} - 1)\max_{\mathbf{x}\in\mathbb{X}}\sqrt{k(\mathbf{x},\mathbf{x})} \end{aligned} \tag{21}$$

where the first line apply Lemma 3. The third line use Lemma 5 and the last line is achieved through Lemma 4. We now try to give the coefficient of L_{μ_t} and L_t, and these coefficients are all based on the smoothness coefficient of Gaussian process. First, according to Theorem 3, with the probability of at least $1 - \frac{\delta}{6}$, for any $i, j = 1, ..., d$, we have

$$L_{ij} = \sup_{\mathbf{x}\in\mathbb{X}}\left|\frac{\partial^2}{\partial x_i\partial x_j}f(\mathbf{x})\right| \le \sqrt{2\log\left(\frac{12d^2}{\delta}\right)}\max_{\mathbf{x}\in\mathbb{X}}\sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + 12\sqrt{6d}\max\left\{\max_{\mathbf{x}\in\mathcal{X}}\sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})}, \sqrt{rL_k^{\partial_{ij}}}\right\} \tag{22}$$

If we carefully choose L_{μ_t} and L_t as

$$L_{\mu_t} = \sqrt{\sum_{i,j} V_{t,ij}^2} \qquad L_t = \sqrt{\sum_{i,j} \tilde{U}_{t,ij}^2}$$

where

$$V_{t,ij} = \sqrt{2\log\left(\frac{\pi^2 t^2 d^2}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + L_{ij}$$

$$\tilde{U}_{t,ij} = L_{ij} + 2\sqrt{2\log\left(\frac{4\pi^2 t^2 d^2}{\delta}\right)} \max_{\mathbf{x}\in\mathbb{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})} + 12\sqrt{6}d \max\left\{\max_{\mathbf{x}\in\mathcal{X}} \sqrt{k^{\partial_{ij}}(\mathbf{x},\mathbf{x})}, \sqrt{rL_k^{\partial_{ij}}}\right\}$$

According to Theorem 5 and 6, this guarantees that

$$\sum_{t=1}^{\infty} P(\mu_{\mathcal{D}_t}(\cdot) \text{ is } L_{\mu_t} - smooth) \geq 1 - \frac{\delta}{6}$$
$$\sum_{t=1}^{\infty} P(U_{L_t}|\mathcal{D}_t) \geq 1 - \frac{\delta}{6}$$

and we also have $P(U_{L_t}^c|D_t) \leq \frac{6\delta}{\pi^2 t^2}$. Thus for the standard derivation term, their approximate error term is tend to be a very small value:

$$c_2 P(f(\cdot) \in U_{L_t}^c|\mathcal{D}_t) + c_1(\frac{1}{P(U_{L_t}|\mathcal{D})} - 1) \max_{\mathbf{x}\in\mathbb{R}^d} \sqrt{k(\mathbf{x},\mathbf{x})} = O\left(\frac{1}{t^2}\right) \tag{23}$$

If we combine Eq. (19), Eq. (20), Eq. (21) and Eq. (23), and we also set $\tilde{\beta}_t = c_1\beta_t$ and $\eta_t = \sqrt{\frac{2}{\pi}} \frac{1}{6\beta_t c_1(L_t + L_{\mu_t})}$, we can obtain an upper bound for $f(\mathbf{x}_{t+1})$, and this upper bound is related to the gradient:

$$\begin{aligned} f(\mathbf{x}_{t+1}) &\leq \mu_{\mathcal{D}_t}(\mathbf{x}_t) + \tilde{\beta}_t\sigma_{\mathcal{D}_t}(\mathbf{x}_t) - \frac{1}{2}\eta_t\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 + \tilde{\beta}_t\sqrt{\frac{\pi}{2}}\eta_t\sqrt{\gamma_t}\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2 + \tilde{\beta}_t O\left(\frac{1}{t^2}\right) \\ &\leq \mu_{\mathcal{D}_t}(\mathbf{x}_t) + \tilde{\beta}_t\sigma_{\mathcal{D}_t}(\mathbf{x}_t) - \frac{1}{2}\eta_t\|\nabla f(\mathbf{x}_t)\|_2^2 + \frac{1}{2}\eta_t(\|\nabla f(\mathbf{x}_t)\|_2^2 - \|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2) \\ &\quad + \tilde{\beta}_t\sqrt{\frac{\pi}{2}}\eta_t\sqrt{\gamma_t}\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2 + \tilde{\beta}_t O\left(\frac{1}{t^2}\right) \end{aligned}$$

Through simple analysis, we have

$$\begin{aligned} \|\nabla f(\mathbf{x}_t)\|_2^2 - \|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2^2 &= (\|\nabla f(\mathbf{x}_t)\|_2 - \|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2)(\|\nabla f(\mathbf{x}_t)\|_2 + \|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2) \\ &= -(\|\nabla f(\mathbf{x}_t)\|_2 - \|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2)^2 + 2(\|\nabla f(\mathbf{x}_t)\|_2 - \|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2)\|\nabla f(\mathbf{x}_t)\|_2 \\ &\leq 2\|\nabla f(\mathbf{x}_t) - \nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2\|\nabla f(\mathbf{x}_t)\|_2 \end{aligned}$$

and

$$\|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t)\|_2 \leq \|\nabla\mu_{\mathcal{D}_t}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|_2 + \|\nabla f(\mathbf{x}_t)\|_2$$

Thus if we consider the Lemma 7, we can transform the previous upper bound into the following result with probability $1 - \frac{\delta}{6}$ for any $t > 1$

$$\begin{aligned} f(\mathbf{x}_{t+1}) \leq \mu_{\mathcal{D}_t}(\mathbf{x}_t) + \tilde{\beta}_t\sigma_{\mathcal{D}_t}(\mathbf{x}_t) - \frac{1}{2}\eta_t\|\nabla f(\mathbf{x}_t)\|_2^2 + 2\tilde{\beta}_t\sqrt{\frac{\pi}{2}}\eta_t\sqrt{\gamma_t}\|\nabla f(\mathbf{x}_t)\|_2 \\ + \tilde{\beta}_t^2\sqrt{\frac{\pi}{2}}\eta_t\gamma_t + \tilde{\beta}_t O\left(\frac{1}{t^2}\right) \end{aligned}$$

Based on the above result, we can further obtain the following results with probability $1 - \frac{\delta}{6}$ for any $t > 1$

$$
\begin{aligned}
f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) &\leq \mu_{\mathcal{D}_t}(\mathbf{x}_t) + \tilde{\beta}_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t) - f(\mathbf{x}_t) - \frac{1}{2}\eta_t \|\nabla f(\mathbf{x}_t)\|_2^2 + 2\tilde{\beta}_t \sqrt{\frac{\pi}{2}} \eta_t \sqrt{\gamma_t} \|\nabla f(\mathbf{x}_t)\|_2 \\
&\quad + \tilde{\beta}_t^2 \sqrt{\frac{\pi}{2}} \eta_t \gamma_t + \tilde{\beta}_t O(\frac{1}{t^2}) \\
&\leq 2\tilde{\beta}_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t) - \frac{1}{2}\eta_t \|\nabla f(\mathbf{x}_t)\|_2^2 + 2\tilde{\beta}_t \sqrt{\frac{\pi}{2}} \eta_t \sqrt{\gamma_t} \|\nabla f(\mathbf{x}_t)\|_2 + \tilde{\beta}_t^2 \sqrt{\frac{\pi}{2}} \eta_t \gamma_t + \tilde{\beta}_t O\left(\frac{1}{t^2}\right)
\end{aligned}
$$

Based on the above inequation, after t steps in MinUCB algorithm, we can have the following result with probability of at least $1 - \delta$:

$$
\begin{aligned}
f(\mathbf{x}_T) - f(\mathbf{x}_0) \leq -\frac{1}{2} \sum_{t=1}^{T} \eta_t \left[\|\nabla f(\mathbf{x}_t)\|_2^2 - 4\tilde{\beta}_t \sqrt{\frac{\pi}{2}} \eta_t \sqrt{\gamma_t} \|\nabla f(\mathbf{x}_t)\|_2 \right] + 2 \sum_{t=1}^{T} \tilde{\beta}_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t) \\
+ \sum_{t=1}^{T} \tilde{\beta}_t^2 \sqrt{\frac{\pi}{2}} \eta_t \gamma_t + O(\log \frac{1}{\delta})
\end{aligned}
$$

By organizing the above results, we can obtain

$$
\begin{aligned}
\frac{1}{2} \sum_{t=1}^{T} \eta_t \left[\|\nabla f(\mathbf{x}_t)\|_2^2 - 4\tilde{\beta}_t \sqrt{\frac{\pi}{2}} \sqrt{\gamma_t} \|\nabla f(\mathbf{x}_t)\|_2 \right] &\leq f(\mathbf{x}_0) - f(\mathbf{x}_T) + 2 \sum_{t=1}^{T} \tilde{\beta}_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t) \\
&\quad + \sum_{t=1}^{T} \tilde{\beta}_t^2 \sqrt{\frac{\pi}{2}} \eta_t \gamma_t + O(\log \frac{1}{\delta}) \\
\frac{1}{2} \sum_{t=1}^{T} \eta_t \left[\|\nabla f(\mathbf{x}_t)\|_2 - 2\tilde{\beta}_t \sqrt{\frac{\pi}{2}} \sqrt{\gamma_t} \right]^2 &\leq V^* + 2 \sum_{t=1}^{T} \tilde{\beta}_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t) + 2\pi \sum_{t=1}^{T} \tilde{\beta}_t^2 \eta_t \gamma_t + O(\log \frac{1}{\delta}) \\
\frac{1}{2} \sum_{t=T/2}^{T} \eta_t \left[\|\nabla f(\mathbf{x}_t)\|_2 - 2\tilde{\beta}_t \sqrt{\frac{\pi}{2}} \sqrt{\gamma_t} \right]^2 &\leq V^* + 2 \sum_{t=1}^{T} \tilde{\beta}_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t) + 2\pi \sum_{t=1}^{T} \tilde{\beta}_t^2 \eta_t \gamma_t + O(\log \frac{1}{\delta})
\end{aligned}
$$

Based on this inequality we can directly give the upper bound for the norm of gradient:

$$
\min_{T/2 \leq t \leq T} \eta_t \left[\|\nabla f(\mathbf{x}_t)\|_2^2 - \tilde{\beta}_t \sqrt{\frac{\pi}{2}} \eta_t \sqrt{\gamma_t} \right]^2 \leq \frac{4}{T} V^* + \frac{8}{T} \sum_{t=1}^{T} \tilde{\beta}_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t) + \frac{4\pi}{T} \sum_{t=1}^{T} \tilde{\beta}_t^2 \eta_t \gamma_t + O(\frac{1}{T} \log \frac{1}{\delta})
$$

$$
\begin{aligned}
\min_{T/2 \leq t \leq T} \|\nabla f(\mathbf{x}_t)\|_2 \leq \frac{1}{\sqrt{\eta_T}} \sqrt{\frac{8}{T} \sum_{t=1}^{T} \tilde{\beta}_t \sigma_{\mathcal{D}_t}(\mathbf{x}_t) + \frac{8\pi}{T} \sum_{t=1}^{T} \tilde{\beta}_t^2 \eta_t \gamma_t + O(\frac{1}{T} \log \frac{1}{\delta})} \qquad (24) \\
+ \frac{1}{\sqrt{\eta_T}} \tilde{\beta}_{T/2} \sqrt{\frac{\pi}{2}} \sqrt{\gamma_{T/2}} \qquad (25)
\end{aligned}
$$

Notice that in each step of MinUCB, we do the repeated sampling at point $\mathbf{x}_t$, we denote $\mathcal{D}_t^1 = \{(\mathbf{x}_t, y_t^1), ..., (\mathbf{x}_t, y_t^{b_t^{(1)}})\}$ as this resampling set. We will have

$$
\sigma_{\mathcal{D}_t}(\mathbf{x}_t) \leq \frac{\sigma}{\sqrt{b_t^{(1)}}}
$$

Through Lemma 6, we can achieve

$$\gamma_t \leq E_{d,k,\sigma}(b_t^{(2)})$$

We combine these two value into Eq. (24), which proves this theorem.

□

Theorem 2. *Suppose f is sampled from a zero mean Gaussian process with a continuously differentiable convariance function $k(\cdot,\cdot)$, then if the kernel is Gaussian kernel or Matérn kernel with $\gamma = 2.5$, and satisfy $\beta_t = \sqrt{2\log\frac{\pi^2 t^2}{\delta}}$, batch size*

$$b_t^{(1)} = \begin{cases} \log^2 t \\ t \\ t^2 \end{cases} \qquad \text{and} \qquad b_t^{(2)} = \begin{cases} d\log^2 t \\ dt \\ dt^2 \end{cases}$$

MinUCB will achieve the convergence rate of

$$\min_{T/2\leq i\leq T} \|\nabla f(x_i)\|_2^2 \leq \begin{cases} O(\sigma d^{\frac{3}{2}} T^{-1} \log^{\frac{3}{2}} \frac{d^2T^2}{\delta}) + O(\sigma d^2) \\ O(\sigma d^2 T^{-\frac{1}{2}} \log^{\frac{5}{2}} \frac{d^2T^2}{\delta}) = O(\sigma d^{\frac{9}{4}} n^{-\frac{1}{4}} \log^{\frac{5}{2}} \frac{dn}{\delta}) \\ O(\sigma d^2 T^{-1} \log^{\frac{5}{2}} \frac{d^2T^2}{\delta})) = O(\sigma d^{\frac{7}{3}} n^{-\frac{1}{3}} \log^{\frac{5}{2}} \frac{dn}{\delta})) \end{cases}$$

Proof. According to the proof of Theorem 1, we have

$$\frac{1}{\eta_t} = O(d\sqrt{\log\frac{t^2d^2}{\delta}} + d^{3/2}) \tag{26}$$

Through Lemma 8 and 9, we can bound $E_{d,k,\sigma}(b_t^{(2)})$ as $E_{d,k,\sigma}(b_t^{(2)}) = O\left(\frac{\sigma\sqrt{b_t^{(2)}}}{d}\right)$. Thus if we combline the result of Theorem 1 and above error bounds, we will proof the results.

□

B.5 Theoretical property of LA-MinUCB

In this section we will provide the theoretical property for our LA-MinUCB algorithm, that LA-MinUCB is Bayesian optimal if there is only one iteration left. Our result is proved in a similar way with the Proposition 2 in Wu et al. [35].

Theorem 3. *If only one iteration is left and we can observe the function value through sampling, then the local exploration in LA-MinUCB is Bayes-optimal among all feasible policies.*

Proof. Suppose that we are given a budget of N_{max} samples, i.e. we may run the algorithm for N iterations. Thus, letting Π be the set of feasible policies π, we can formulate our problem as follows:

$$\inf_{\pi\in\Pi} \mathbb{E}^{\pi}\left[\min_{\mathbf{x}\in\mathbb{X}} \mu_{\mathcal{D}_N}(\mathbf{x}) + \beta_N \sigma_{\mathcal{D}_N}(\mathbf{x})\right]$$

We analyze this problem under the DP framework. We define our state space as $S^n = (\mu_{\mathcal{D}_n}(\cdot), k_{\mathcal{D}_n}(\cdot,\cdot))$ after iteration n as it completely characterizes our belief on f. Under the DP framework, we will define the value function V^n as follows:

$$V^n(s) := \inf_{\pi\in\Pi} \mathbb{E}^{\pi}\left[\min_{\mathbf{x}\in\mathbb{X}} \mu_{\mathcal{D}_N}(\mathbf{x}) + \beta_N \sigma_{\mathcal{D}_N}(\mathbf{x}) | S^n = s\right]$$

□

for every $s = (\mu, k)$. The Bellman equation tells us that the value function can be written recursively as

$$V^n(s) = \min_{\mathbf{Z}\in\mathbb{R}^{b_n}\times d} Q^n(s, \mathbf{Z})$$

where

$$Q^n(s, \mathbf{Z}) = \mathbb{E}\left[V^{n+1}(S^{n+1})|S^{n+1}, \mathbf{X} = \mathbf{Z}\right]$$

and $\mathbf{X}$ is the new input in the $n+1^{th}$ step. At the same time, we also know that any policy π^* whose decisions satisfy

$$\mathbf{Z}^{\pi^*,n}(s) \in \arg\min_{\mathbf{Z}\in\mathbb{R}^{b_n\times d}} Q^n(s, \mathbf{Z}) \tag{27}$$

is optimal. If we were to stop at iteration $n+1$, then $V^{n+1}(S^{n+1}) = \min_{\mathbf{x}\in\mathbb{X}} \mu_{\mathcal{D}_{n+1}}(\mathbf{x}) + \beta_{n+1}\sigma_{\mathcal{D}_{n+1}}(\mathbf{x})$ and Eq. (27) reduces to

$$\mathbf{Z}^{\pi^*,n}(s) \in \arg\min_{\mathbf{Z}\in\mathbb{R}^{b_n\times d}} \mathbb{E}\left[\min_{\mathbf{x}\in\mathbb{X}} \mu_{\mathcal{D}_{n+1}}(\mathbf{x}) + \beta_{n+1}\sigma_{\mathcal{D}_{n+1}}(\mathbf{x})|S^n = s, \mathbf{X} = \mathbf{Z}\right]$$

which is exactly the result of local exploration acquisition function of LA-MinUCB. This proves that the local exploration acquisition function in LA-MinUCB is one-step Bayes-optimal.

C Additional experimental results

C.1 Experiements on other real-world objective functions and methods

We further evaluate our methods on other real-world objective functions, which are the same as Nguyen et al.[23]. The initial pair of functions deals with inverse challenges typically found in physics and engineering disciplines. The first one is related to electrical engineering, where the goal is to optimize the alignment of a theoretical model of an electronic circuit with actual data. This involves adjusting nine parameters, with a limit of 500 evaluations for the optimization process. The second function addresses a cosmological issue[24], where the objective is to fine-tune a cosmological model or physical simulator to match observational data from the cosmos. The focus is on maximizing the log likelihood of the model, which is influenced by various physical constants. We follow the setting in Eriksson et al.[7], which presents a harder optimization problem with 12 parameters and much larger bounds, and set the budget at 2000 evaluations. The third function concerns rover trajectory planning [33]. It involves optimizing the placement of 100 points in a 2D space to plot the rover's trajectory and reduce costs, creating a 200-dimensional optimization problem with a budget of 1000 function evaluations. The experimental results are shown in Fig 4. This experiment shows that LA-MinUCB remains the most competitive method, with its performance showing a significant improvement compared to other methods.

To demonstrate the performance difference between our method and the original UCB, we conducted a simple experiment on the synthetic functions introduced in Section 8.1, which is shown in Fig 5. The parameters of the traditional UCB method were consistent with ours, with the UCB parameter β set to 3. Under these settings, we compared the performance of these methods. The experiment showed that the performance of the traditional UCB was very limited across these three types of problems, struggling to find the maximum value of the function, and this issue became more severe as the data dimension increased. In contrast, our method exhibited good scalability with respect to dimensions, achieving favorable results across various dimensions. This indicates that through our proposed methods, we have transformed the traditional UCB concept into local optimization, achieving very good results and making a significant difference from the original UCB performance.

C.2 Ablation study

In the ablation study, we examine the impact of the core parameter β on MinUCB and LA-MinUCB. We consider three values for β, which are 1, 3, and 5. When β is set to 1, it indicates that the algorithm adopts a more aggressive strategy, allowing it to search for points with higher uncertainty during the descent step. Conversely, when β is set to 5, the algorithm tends to be more conservative, favoring the search for points that are very likely to have function values less than the current best point. We list the experimental results on synthetic function defined in Section 8.1 under these three β values in Figure 6,7 and 8. The results show that when β is small, both MinUCB and LA-MinUCB are more likely to achieve better results in the initial search phase. This is because a smaller beta value allows for a broader search range, increasing the chances of finding relatively good results early on. However, in the later stages of finding the optimal point, it may be weaker than when β is larger. This is because an overly aggressive strategy can cause the algorithm to hover around the optimal

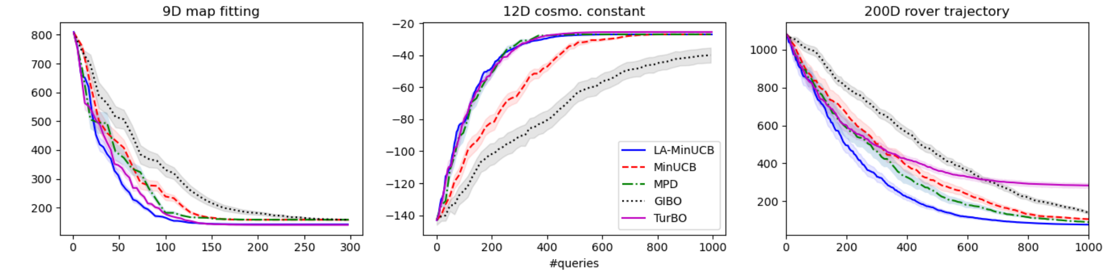

Figure 4: Progressive objective values observed on real-world tasks. LA-MinUCB is competitive against other baselines on all tasks.

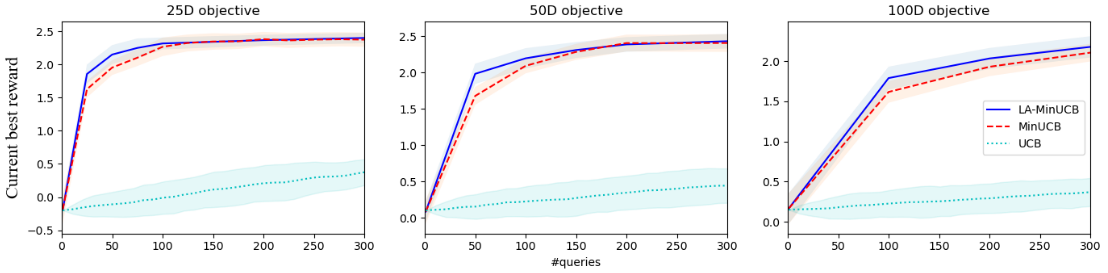

Figure 5: Progressive objective values observed on synthetic function. Our methods are much better than traditional UCB methods

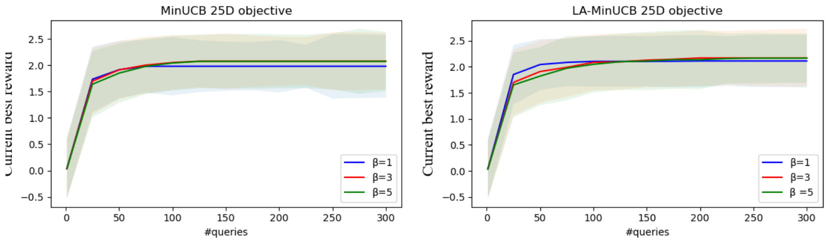

Figure 6: Progressive objective values observed on synthetic function when $D = 25$.

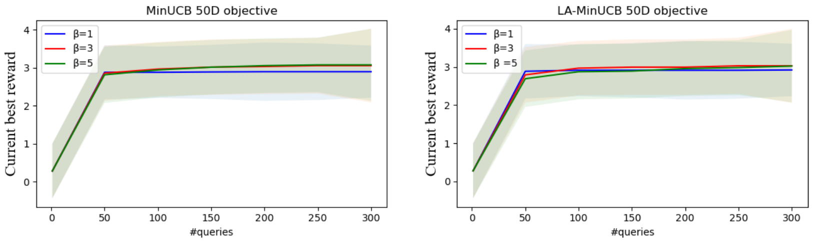

Figure 7: Progressive objective values observed on synthetic function when $D = 50$.

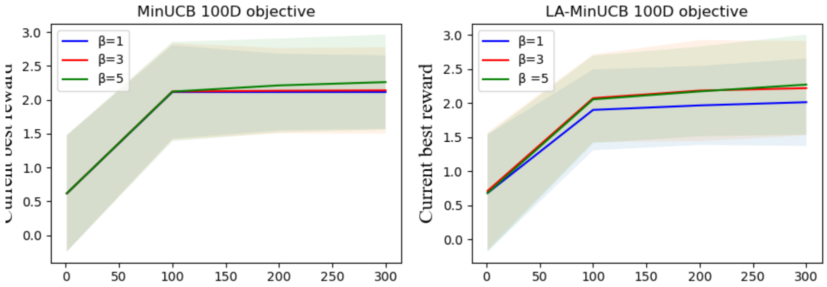

Figure 8: Progressive objective values observed on synthetic function when $D = 100$.

point without getting close. This can also be seen from the Theorem 2, which states that a larger β is a necessary factor for the algorithm to approach the extreme point. When β is set to 3, it represents a more balanced outcome, achieving a good convergence rate while the optimal point is also close to that when β is set to 5, making it a better choice.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes, please refer to our Section 9.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

Justification: Our Appendix B provide the detailed proof of our algorithms.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: Please refer to experiment settings in Section 8. The codes for the proposed algorithm and experiments will also be made publicly available upon publication.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The codes for the proposed algorithm and experiments will also be made publicly available upon publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

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

   Answer: [No]

   Justification: Our experiment in the article is completely consistent with Nguyen et al. [23], and specific details can be found in their article.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [Yes]

   Justification: Please refer to our experiments in Section 8.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: Please refer to our experiments in Section 8.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: This study purely contributes to the technical advancement of local Bayesian optimization algorithm and does not have any societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Please refer to our experiment in Section 8.

    Guidelines:

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

13. **New Assets**

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

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

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