Improving sample efficiency of high dimensional Bayesian optimization with MCMC on approximated posterior ratio

Anonymous Authors

Anonymous Affiliations

Abstract

Efficient optimization in high dimensional space is a fundamental challenge in almost all scientific disciplines. The curse of dimensionality significantly deteriorates the performance of sequential optimization methods. Previous approaches partition and constrain the optimization problem on small regions to prevent overemphasized exploration and inefficient exploitation. However, the complexity of keeping Gaussian process posteriors often limits the number of candidate points and therefore reduces the efficiency of optimization. We propose a Markov chain based sampling algorithm MCMC-BO to overcome this difficulty by moving the candidate points towards more reasonable positions. MCMC-BO can serve as an overarching algorithm with only two additional hyper-parameters by using existing Bayesian optimization methods. We provide theoretical guarantees on its convergence under Gaussian process Thompson sampling setting. We also show in experiments that MCMC-BO outperforms state-of-the-art methods in high dimensional sequential optimization and reinforcement learning benchmarks.

Introduction

Black-box function optimization is an essential task in machine learning for neural network hyper-parameter tuning, reinforcement learning, etc. It also has wide applications in physical simulation, chemical design, and biological discovery. There is often a lack of gradient information in these non-convex sequential optimization problems. Bayesian optimization (BO) is a popular sampling based optimization approach with versatility in black-box function optimization. It has been successfully applied in problems such as online learning and sequential decision making. BO builds a surrogate model for modeling the objective function and optimizes the acquisition function to propose new samples.

Similar to many other numerical problems, Bayesian optimization algorithms are also susceptible to the curse of dimensionality. The search space would grow exponentially as function dimension increases and become intractable under limited computation budget. Common acquisition functions also tend to over-explore the uncertainty boundary region and lacks exploitation in high dimensional input space(Oh, Gavves, and Welling 2018). Recent developments in high dimensional BO include constructing trust regions and space partition to improve the probability of sample in the promising regions, which effectively ameliorates the problem of over-exploration(Eriksson et al. 2019; Wang, Fonseca, and Tian 2020). To evaluate acquisition function on a continuous domain, these methods often discretize the search space using Sobolev sequence(Sobol' 1967). This kind of discretization is inadequate on high dimensional space as the size of the discretization set are limited, hindering the exploitation of potential good regions.

To improve optimization performance in high-dimensional spaces, we adopt Markov Chain Monte Carlo (MCMC) for BO, which is widely used and can effectively sample from high-dimensional posterior distributions. We propose MCMC-BO, which transits candidate points from original positions towards the approximated stationary distribution of Thompson sampling. Tracking only batch size number of points during transitions, MCMC-BO significantly lowers the vast storage need of candidate points. Our proposed method can be easily generalized to different circumstances with only two additional hyper-parameters: Brownian motion's noise and number of transitions. With theoretical performance guarantees, MCMC-BO can serve as an overarching algorithm linked to any existing BO method on continuous problems.

Contributions. 1) We propose MCMC-BO, a Bayesian optimization algorithm which enables better local optimization in high-dimensional problems with MCMC. 2) We provide theoretical guarantees for the regret bound of MCMC-BO. To our knowledge, this is the first regret bound on high-dimensional Bayesian optimization problem which can deal with the scaling of dimensions on candidate points and avoid the overuse of memory. 3) We experimentally show that MCMC-BO outperforms other strong baselines on various high-dimensional tasks. We also provide a baseline for implementing MCMC-BO .

Related Work

The optimization of black-box functions has been broadly used in many scenarios, such as hyper-parameter tuning(Snoek, Larochelle, and Adams 2012) and experimental design(Hernández-Lobato et al. 2017). These kinds of problems can be formalized under the Bayesian optimiza-

tion framework (Shahriari et al. 2015; Frazier 2018). While conventional BO algorithms are limited to less than 15 parameters and a few thousand observations, there are more and more attempts to scale BO to higher dimensional cases and sample more points with lower computation budgets. Current efforts include varying or substituting the use of GP in the process of optimization, assuming low dimensional data or function structures, or preventing explorations near domain boundary.

The scalability of the algorithm can be improved by using either variants of Gaussian process (GP) or other different methods. Using sparse GP, we can compute function evaluation on large number of points(Seeger, Williams, and Lawrence 2003; Snelson and Ghahramani 2005; Hensman, Fusi, and Lawrence 2013). Several algorithms also replace the use of GP(Rasmussen 2003) with other surrogate models such as Random Forest(Hutter, Hoos, and Leyton-Brown 2011), Bayesian Neural Network(Springenberg et al. 2016; Snoek et al. 2015; Hernández-Lobato et al. 2017), and tree Parzen estimator(Bergstra et al. 2011; Falkner, Klein, and Hutter 2018) to track large number of observations.

A popular line of research assumes the existence of some underlying low dimensional structure and transforms high dimensional space into a lower dimensional subspace in which BO is feasible. Several works utilizing random embeddings can effectively solve problems with the inputs having millions of dimensions(Wang et al. 2016; Letham et al. 2020). Nayebi et al. provides a theoretical hashing-based framework, HesBO, for BO with subspace embeddings(Nayebi, Munteanu, and Poloczek 2019). Beyond linear embeddings, deep generative model can be used to learn a non-linear mapping between high dimensional input space and low-dimensional subspace(Lu et al. 2018; Tripp, Daxberger, and Hernández-Lobato 2020).

Another line of research relies on the additive structure of the objective function, which decomposes input space into low dimensional components and therefore reduces the dimension of sub-problems(Kandasamy, Schneider, and Póczos 2015; Gardner et al. 2017; Mutny and Krause 2018; Wang et al. 2018; Kirschner et al. 2019).

Many works also try to prevent BO algorithms from over-exploring uncertain points near the domain boundary. Some of them introduce strong priors or dimension dropout to relieve this issue(Eriksson and Jankowiak 2021; Li et al. 2018). Oh et al. introduces a cylindrical kernel to allocate more sample points in the center region(Oh, Gavves, and Welling 2018). TuRBO uses dynamic trust regions to restrict samples within hyper-cubes centered around the current best points(Eriksson et al. 2019). LA-MCTS recursively partitions the input space using Monte Carlo Tree Search (MCTS), allowing BO algorithms to optimize on promising regions(Wang, Fonseca, and Tian 2020).

Besides Bayesian optimization, evolutionary algorithms (EA) are common approaches to tackle high dimensional function optimization(Jin and Branke 2005). CMA-ES is a popular EA algorithm which adaptively adjusts the covariance matrix to generate new samples(Hansen, Müller, and Koumoutsakos 2003). Shiwa is a rule-based method that automatically selects suitable EA algorithms according to the property of problems, achieving better performance than other single EA methods on many tasks(Liu et al. 2020).

Methods

Black-box optimization problems aims to find

$$\mathbf{x}^* \in \Omega \text{ such that } f(\mathbf{x}^*) \geq f(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega \tag{1}$$

where $f : \Omega \to \mathbb{R}$ and $\Omega = [0, 1]^d$. The observation may come with Gaussian noise that $y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$. Since the objective function f is unknown, approximating f is a crucial step towards optimization.

Bayesian optimization methods are standard paradigms for none-gradient black-box problems. Like many other methods, it faces the difficulty of scaling to higher dimensions. The number of samples required for maintaining the same level of granularity grows exponentially with dimension. Moreover, with minimal assumptions (e.g. Lipschitz continuity) on the structure of f to allow for generalization, solving the problem could take time that grows exponentially with the number of input dimensions.

In Bayesian optimization, GP is among the most prevalent method. GP produces estimates for function values together with appropriate confidence intervals to quantify uncertainty, which can be employed to balance between exploration and exploitation. The goal of optimization is to sample points near the optimum while effectively explore the sample space. Properly choosing sampling points becomes a much more critical issue when working in a high dimensional space, in order to successfully employ GP methods without sampling a large amount of data points that both incur regret and waste computational resources. BO first constructs a surrogate model to give a probabilistic guess with small number of samples, then updates the chosen points based evaluations or further information. Following the principle of optimism in the face of uncertainty, one line of work, most notably Thompson sampling (TS), models the reward using probabilistic methods.

Batch optimization attempts to pick the best m (batch size) points for further evaluation. For high-dimensional space, full exploration demands an unrealistic amount of computational resource and time. Therefore, the ability to perform adequate exploitation on optimal points determines the final performance. Previous algorithms that attempts to optimize on high-dimensional spaces include TuRBO and LA-MCTS. TuRBO limits the candidate set to a small box around best points and decides to expand or shrink the box based on later evaluations. LA-MCTS partition the region with a constructed tree, and MCTS recommends promising regions to improves the performance of the algorithm. Both the methods works on how to focus on promising regions. However, in comparison to finding boundaries for the best points, we argue that the ability to congregate the candidate points around the optimum could be more important and also deserves attention. Discretization on continuous domains makes practical implementations plausible, but maintaining discretized grids with high precision is costly. To illustrate, GP-TS (Chowdhury and Gopalan 2017) requires that, to achieve the theoretical upper bound on regret, the size of round t's discretization set should at least be

$|D_t| = \left(BLrdt^2\right)^d$, where B is bound of f's RKHS norm, L is Lipchitz constant, r is the region length, and d the dimension. Even after partitioning with bifurcated tree with tiny r (the method proposed by LA-MCTS), achieving such a vast candidate set size on high dimensional spaces is unrealistic due to limitations of computation storage. The trade of between regret and storage urges the need for improvements in current methods.

We adopt MCMC methods to propose a novel algorithm, MCMC-BO , to effectively evaluate more points while having the ability to maintain a dense, time-varying, discretization near promising set of regions for optimization.

Gaussian Process

In Bayesian Optimization, the unknown objective f is viewed as a probability distribution, and Gaussian process (GP) utilizes such idea to stochastically model the unknown f. GP can be determined by it's mean $\mu(\cdot)$ and kernel $k(\cdot,\cdot)$. The prior distribution of $f(x)$ is assumed to be $\mathcal{N}(0, k(x,x))$. The sample points $A_T := [x_1, \cdots, x_T]$ and the observations $[y_1, \cdots y_T]$ follows the multivariate distribution $\mathcal{N}(0, K_T + \sigma^2 I)$, where $K_T = [k(x,x')]_{x,x' \in A_T}$. Therefore, for any $x \in \Omega$, we can obtain $\mathbb{P}(x \mid x_1, \cdots, x_T)$. The posterior distribution over f is thus Gaussian with mean $\mu(x)$ and covariance $k_T(x,x')$ that satisfy:

$$\begin{aligned}
\mu_T(x) &= k_T(x)^T (K_T + \sigma^2 I)^{-1} y_T \\
k_T(x,x') &= k(x,x') - k_T(x)^T (K_T + \sigma^2 I)^{-1} k_T(x') \\
\sigma_T^2(x) &= k_T(x,x'),
\end{aligned}$$

where $k_T(x) = [k(x_i, x)]_{i=1:T}^T$.

Thompson Sampling

Thompson Sampling (TS), often employed in Multi-armed Bandits (MAB) problems, employs an Bayesian approach by assigning a prior distribution for the reward estimate of each arm and updating the posterior through repetitive plays of wisely-chosen arms. In every round, the algorithm estimate the reward of every arm by sampling from their prior distributions and plays the arm with the largest estimate. Thus, exploitation of optimal arms is achieved by heuristically playing the optimal arm and exploration guaranteed by the intrinsic uncertainty of probabilistic sampling. For Gaussian arms with dependencies among each other, it is natural to use GP to approximate the underlying reward distribution. Chowdhury and Gopalan gave a theoretical analysis for the algorithm GP-TS, which combines GP and TS to optimize stochastic MAB problems (Chowdhury and Gopalan 2017). With an evolving GP, at each round t, GP-TS samples a function f_t from GP$(\mu_t(\cdot), k_t(\cdot,\cdot))$ and choose the action x_t that maximizes f_t.

However, naively employing GP-TS in higher-dimensional space to minimize an underlying f could be problematic. The average spacing between a fixed number of data points scales with the dimension, which leads to a higher variance in the GP and encourages over-exploration without sufficient exploitation and optimization over the promising region.

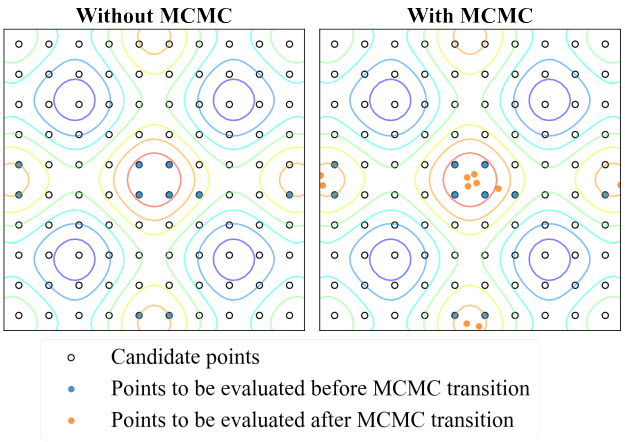


Figure 1: Illustration of MCMC-BO. The contours are 2d Rastrigin function. (**Left**): BO algorithms propose points to be sampled. The optimization performance is restricted by insufficient discretization. (**Right**): Points are adjusted by MCMC-BO , reaching regions with higher value.

Markov Chain Monte Carlo Method

To solve the problem of over-exploring the high-dimensional sample space, we can utilize Markov Chain Monte Carlo (MCMC) methods that allows for a more efficient and effect way of choosing and transitioning between sampling points.

Metropolis-Hastings (MH) is a MCMC sampling algorithm that can sample from a target distribution $\pi(x),\ x \in \mathcal{X}$ known up to a constant, i.e. when we have knowledge of $\pi_d(x) = c \cdot \pi(x)$ (Metropolis et al. 1953). The constant c is often the normalizing constant for the distribution $\pi(x)$ and the MH algorithm is especially useful when finding explicit solution to the integral is difficult. We also need to choose an appropriate proposal distribution $q(\cdot, x)$ that has support on all of the support of $\pi(\cdot)$; the variance of $q(\cdot, x)$ influences tendencies to exploit or explore, depending on different target distributions. From $q(\cdot, x)$, the algorithm samples a candidate point $y \in \mathcal{X}$ given the current value x according to $q(y, x)$. Then the Markov Chain transitions to y with acceptance probability $\alpha(x,y) = \min\left\{1, \frac{\pi_d(y)q(y,x)}{\pi_d(x)q(x,y)}\right\}$.

Therefore, the transition probability $\alpha(x,y)$ naturally induces a transition kernel for the Markov Chain given by

$$K_{MH}(y,x) = q(y,x)\alpha(x,y) + \delta_x(y) \int_{\mathcal{X}} q(y,x)\big(1 - \alpha(y,x)\big)dy.$$

It can be verified that $K_{MH}(\cdot,\cdot)$ is reversible w.r.t. $\pi(\cdot)$, i.e.

$$\pi(x)K_{MH}(x,y) = \pi(y)K_{MH}(y,x),$$

which guarantees that the MH algorithm eventually converges to a stationary distribution that is exactly $\pi(\cdot)$ (Andrieu et al. 2003).

Approximation of TS with MH algorithm

Thompson sampling method samples from the posterior of GP Regression and picks the point with the best result.

From another perspective, Thompson sampling gives each candidate point x_i a probability $P(x_i) = P(f_t(x_i) \geq f_t(x_j)), i \neq j$. The calculation of the probability of each point is equivalent to the calculation of the CDF of multivariate normal distribution, which is indeed intractable when dimension is greater than 2. On the other hand, sampling from a multivariate normal distribution require its Cholesky factorization. The computation cost of factorizing the covariance matrix $k_T(x^*, x^*) = k(x, x) - k_T(x*)^T(K_T + \sigma^2 I)^{-1}k_T(x^*)$ of the newest fashion is $O(m^2 * n + n^3)$(Stanton et al. 2021). However. High dimension space could need numerous points lead to large matrix and slow prediction.

Algorithm 1: MCMC-BO

Input Initial dataset $\mathcal{D}_0$, batch size m, MCMC transition number N, transition noise $\boldsymbol{\sigma}$
for $t = 0, 1, \cdots$ **do**
  Update posterior distribution on f using $\mathcal{D}_t$
  Create discretized candidate points $\boldsymbol{x}_t^0$ from continuous search domain
  {MCMC transition start}
  for $i = 0$ to $N - 1$ **do**
    for $k = 0$ to $m - 1$ **do**
      Sample $u \sim$ Unif$[0, 1]$
      Sample $x_{tk}^{i+1} \sim x_{tk}^i + \mathcal{N}(0, \boldsymbol{\sigma})$
      if $u \geq \min\left\{1, \frac{P(f_t(x_{tk}^{i+1})) \geq P(f_t(x_{tk}^i))q(x_{tk}^{i+1}|x_{tk}^i)}{P(f_t(x_{tk}^i)) \geq P(f_t(x_{tk}^{i+1}))q(x_{tk}^i|x_{tk}^{i+1})}\right\}$ **then**
        $x_{tk}^{i+1} \leftarrow x_{tk}^i$ // Reject the transition
      end if
    end for
  end for
  {MCMC transition end}
  Observe $\boldsymbol{y}_t = f(\boldsymbol{x}_t^N)$
  $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup (\boldsymbol{x}_t^N, \boldsymbol{y}_t)$
end for

Markov Chain Monte Carlo are useful when sampling from intractable distribution, and the MH algorithm is one of the fundamental algorithms in this field. According to traditional MH method, the accept rate is $\alpha = \min\left\{1, \frac{P(f_t(x_p) \geq f_t(x), \forall x \in D_t)q(x_p|x_o)}{P(f_t(x_p) \geq f_t(x), \forall x \in D_t)q(x_o|x_p)}\right\}$, where x_p stands for the proposal point, x_o stands for the original point and $q(\cdot \mid \cdot)$ stands for the transition kernel. As we describe above, this is still intractable, so we make an approximation.

Suppose $x = (x_o, x_p)$ are two candidate points after n samples denoted by history $\mathcal{H} = \{(x_t, y_t)\}_{t=1}^n$, we can view the function values $y := (y_p, y_o) = (f_t(x_p), f_t(x_o))$ associated to selected points (x_p, x_o) as a jointly-Gaussian random vector with distribution stated as the GP regression stated above. From linear transformations of Gaussian random vector, let $c^T = [1 \quad -1]$ and $K_n = [k(x_{1:n}, x_{1:n})] \in \mathbb{R}^{n \times n}$., we have that $y_p - y_o \sim \mathcal{N}\left(c^T\mu, c^T\Sigma c\right)$, where $c^T\mu = [k(x_p, x_{1:n}) - k(x_o, x_{1:n})]^T(K_n + \sigma^2 I)^{-1}y_{1:n}$ and $c^T\Sigma c = [k(x_p, x_p) + k(x_o, x_o) - 2k(x_p, x_o)] - (ck_n(x))^T(K_n + \sigma^2 I)^{-1}(ck_n(x))^T$ Since $c^T\mu, c^T\Sigma c \in \mathbb{R}$, we have that $P(y_p - y_o > 0|\mathcal{H}) = \Phi\left(\frac{c^T\mu}{c^T\Sigma c}\right)$ (here $\Phi(\cdot)$ denotes the CDF of the standard normal).Therefore

$$\alpha = \min\left\{1, \frac{\Phi\left(\frac{c^T\mu}{c^T\Sigma c}\right)q(x_p|x_o)}{(1 - \Phi\left(\frac{c^T\mu}{c^T\Sigma c}\right))q(x_o|x_p)}\right\}$$

.

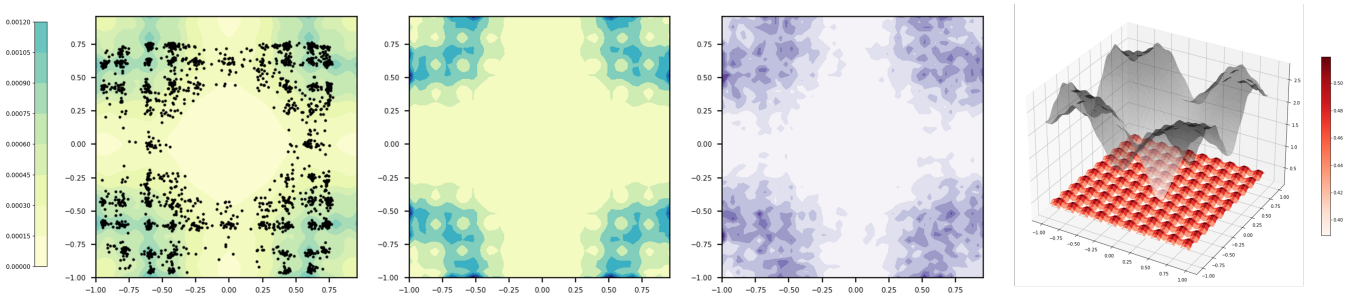
A demonstration of MCMC-BO is shown in Algorithm 1. Our proposed method is compatible with any existing BO algorithms over continuous domain (see Appendix). MCMC-BO prepares a batch of m candidate points x_{tk}^i each round either from direct discretization or points to be executed from other algorithms, where t stands for the round number, $k = 1, 2, 3, \cdots, m$, and i stands for transition times. Then with a proposed transition kernel, often defined as the Brownian motion: $x^{i+1} - x^i \sim \mathcal{N}(0, \boldsymbol{\sigma})$, we generate m pairs of points x_{tk}^i, x_{tk}^{i+1}. We accept or decline the transition with the ratio stated above. The random walk of Markov chain enables dense discretization of continuous space on more optimal regions, as illustrated in Fig 1.

Based on this we can also derive the gibbs sampling version of MCMC-BO. As gibbs sampling is a special version of MH algorithm. It transit the sample on only one dimension at a time. With $Q\left(x'_d, x_{-d} \mid x\right) = x'_i \sim U(x_d)$ where x_d, x^{-d} stands for the d^{th} dimension of the sample and x^{-d} stands for the rest dimension of the sample. Therefore, $\alpha = \frac{p\left(x'_d, x_{-d}\right)Q\left(x_d, x_{-d}|x'_d, x_{-d}\right)}{p\left(x_d, x_{-d}\right)Q\left(x'_d, x_{-d}|x_d, x_{-d}\right)} = 1$ Gibbs sampling based algorithm fits more properly for vanilla Bayesian optimization with randomized intialization like Soblev sequence. It transits the candidate points on one dimension at each round with Thompson sampling.

Theoretical Analysis

The key contribution of our theoretical analysis lies in three aspects. Firstly, the original GP-TS setting requires a discretization set D_t of the function Domain D. However, the size of the set $|D_t|$ goes exponentially with the number of dimensions. We lift the set size requirement in our analysis and are still able to obtain the same upper bound of GP-TS with only a difference in the constant term. Moreover, the inversion of the covariance matrix required by Thompson sampling is of at least $O(t^4)$ complexity, where t stands for number of rounds. Our algorithm abandons this requirement and substitute with the need of number of MCMC's transition times. Lastly, we prove that the approximation of original Thompson sampling does not satisfy MH's reversibility, but it still has a stationary distribution which can be bounded.

The following notation is used in the proofs. Let $\{x_t\}_{t=1}^{\infty}$ be an $\mathbb{R}^d$-valued discrete time stochastic process predictable with respect to the filtration $\{\mathcal{F}_t\}_{t=0}^{\infty}$, i.e., x_t is $\mathcal{F}_{t-1}$,$\forall t \geq 1$, Let$\{\varepsilon_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that for some $R \geq 0$ and for all $t \geq 1, \varepsilon_t$ is (a) $\mathcal{F}_t$-measurable, and (b) R-sub-Gaussian conditionally on $\mathcal{F}_{t-1}$.Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a symmetric, positive-semidefinite kernel.We use the notion of regret as the evaluation metric for the performance of our algorithm. Denote $x^* = \text{argmax}_{x \in D} f(x)$, the cumulative regret over the horizon T of our algorithm



(a) (b) (c) (d)

Figure 2: The figures are constructed from a 50×50 discretization of $D = [-1, 1]^2$. **(a)**: The stationary distribution of MCMC-BO and the congregated points obtained after convergence of the transition process from current GP information, **(b)**: Thompson sampling distribution simulated using Monte Carlo, **(c)**: standard deviation of TS distribution over 10 trials of 10^6 samples, and **(d)**: GP posterior with surfaces being μ and Σ on which MCMC-BO transitions are performed.

is denoted by $R_T = \sum_{t=1}^{T} r_t$, where $r_t = f(x^*) - f(x_t)$. We also impose smoothness assumptions on f in the Reproducing Kernel Hilbert Space (RKHS) such that for $f \in H_k(D)$, $||f||_k \leq B$ and f satisfy a Lipschitz continuous with constant $B \cdot L$. Here $H_k(D)$ is the corresponding RKHS. In our analysis, we work with a discretization D_t with size $|D_t| = \left(BLrdt^2\right)^d$ such that for every $x \in D$, $|f(x) - f([x]_t)| \leq \|f\|_k L \|x - [x]_t\|_1 \leq 1/t^2$,, where $L = \sup_{x \in D} \sup_{j \in [d]} \left(\frac{\partial^2 k(p,q)}{\partial p_j \partial q_j}\Big|_{p=q=x} \right)^{1/2}$. Therefore, we can bound the deviations of available points from this time-varying discretization to the true optimum which may not be obtainable due to such discretization.

We prove the regret bound of MCMC-BO following a similar approach as in the proofs for TS, GP-UCB, and GP-TS (Thompson 1933; Srinivas et al. 2009; Chowdhury and Gopalan 2017). The detailed proof is attached in the Appendix.

We first give formal definition of Markov chain and the transition kernel.

Definition 1. *From measure theory we recall the concept of a kernel. Consider two measure spaces* $(\Omega, \mathcal{A})$ *and* $(\Omega', \mathcal{A}')$. *The function*

$$K : \Omega \times \mathcal{A}' \quad \rightarrow \quad [0, \infty) \tag{2}$$

is a kernel from $(\Omega, \mathcal{A})$ *to* $(\Omega', \mathcal{A}')$ *if*

- $\omega \mapsto K(\omega, A')$ *is* $\mathcal{A}$*-measurable* $\quad \forall \quad A' \in \mathcal{A}'$
- $A' \mapsto K(\omega, A')$ *is a measure on* $\mathcal{A}' \quad \forall \quad \omega \in \Omega$.

If $K(\omega, \Omega') = 1$, *the kernel is a so-called Markov Kernel.*

In a similar fashion to the MH algorithm, which generally can be described by a transition kernel $Q(x, A)$, $x \in \mathbb{R}^d$, $A \in \mathbb{B}(X)$ corresponding to the proposal density $q(x, y)$ (given by kernel $Q(x, dy)$), the acceptance probability $\alpha(x, y)$ of MCMC-BO is given by $\alpha(x, y) = \min\left\{1, \frac{\pi_d(y)q(y,x)}{\pi_d(x)q(x,y)}\right\}$. Denote $\Phi = \{\Phi_n, \ n \in \mathbb{Z}_{>0}\}$ as the Markov chain induced by MCMC-BO and $\pi(x)$ as the stationary distribution.

Theorem 1. *For the proposed approximated transition method, it is not a reversible Markov chain, but it still has a stationary distribution* $\pi(x)$.

In Fig 2 we demonstrate a 2D demo of the stationary distribution of MCMC-BO and compared it with TS setting. As the size of discretization in higher dimensional space can easily blow up, we only track a batch of m points at a time, and do not require the inversion matrix of the points's covariance. With enough transition times, we can still locate the points in the promising region with high probability. Therefore, we can achieve a similar bound on the difference between $f(x)$ and $f([x]_t)$ when x is near the optimum by the increase of MCMC-BO's transition times.

Lemma 1. *For any stationary distribution from the* MCMC-BO*'s chain* $\pi(x), \exists n_\pi, \forall n \geq n_\pi$ *we can reach a close distribution* $\|P^n(x, \cdot) - \pi\| \leq \epsilon$

Regret increases when the approximation function can no longer accurately represent the underlying f. Thus, we hope that the function values of selected points x_t and the approximation at each time step t are not too far away. We define a benign event $E^f(t)$ and a benigh set G_t here.

Definition 2. *Define* $E^f(t)$ *as the event that for all* $x \in D$,

$$|\mu_{t-1}(x) - f(x)| \leq v_t \sigma_{t-1}(x),$$

where $\tilde{c}_t = \sqrt{4 \ln t + 2d \ln\left(BLrdt^2\right)}$ *and* $c_t = v_t\left(1 + \tilde{c}_t\right)$

Definition 3. *Define the set of points with better estimated mean in discretization* D_t *at round* t *as*

$$G_t := \{x \in D_t : \mu_{t-1}(x) \geq f\left([x^\star]_t\right) - v_t \tilde{c}_t \sigma_{t-1}(x)\}$$

This bound measures the difference between function values at the closet point to $x^\star$ *in* D_t *and at the* $t-1$*'s round GP's mean value of x.*

Theorem 2. *For discretization* D_t *stationary distribution* π *and superior set*

$$\Xi_t := \{\mu_{t-1}(x) \geq f(x^\star) - \epsilon\}$$

we have that $\forall \epsilon > 0, \exists \ t$, *s.t.* $\pi(\Xi_t) > q$ *where* q *is a constant from the transition matrix.*

Lemma 2. *For any history $\mathcal{H}'_{t-1}$ such that $E^f(t)$ is true,*

$$\mathbb{P}\left[\Xi_t \in G_t\right] \geq 1 - 1/t^2.$$

Theorem 3. *For any filtration $\mathcal{F}'_{t-1}$ such that $E^f(t)$ is true,* $\mathbb{P}\left[x_t \in G_t \mid \mathcal{F}'_{t-1}\right] \geq p - 2/t^2$

We organize the reward with a regular form into two part. From the choice of discretization sets D_t, the instantaneous regret at round t is given by $[x^\star]_t$ which is the closest point to $x^\star$ in D_t. And $r_t = f(x^\star) - f([x^\star]_t) + f([x^\star]_t) - f(x_t) \leq \frac{1}{t^2} + \Delta_t(x_t)$ Now at each round t, after an action is chosen, MCMC-BO updates the GP based on evaluation of $f(x_t)$. However, if we play a suboptimal arm, the regret suffered can be much higher than the improvement of our knowledge. To overcome this difficulty, at any round t, we divide the arms (in the present discretization D_t) into two groups: G_t and otherwise. By showing that the probability of playing an arm from $D_t \backslash G_t$ is small, the regret from undesirable arm is then bounded.

Theorem 4. *Let $\delta \in (0,1), D \subset [0,r]^d$ be compact and convex, $\|f\|_k \leq B$ and $\{\varepsilon_t\}_t$ a conditionally R-sub-Gaussian sequence. Running GP-TS for a function f lying in the RKHS $H_k(D)$ and with decision sets D_t chosen as above, with probability at least $1-\delta$, the regret of* MCMC-BO *satisfies* $\tilde{O}\left(\gamma_T\sqrt{dT}\right)$

Experiments

In this section, we evaluate MCMC-BO on both high-dimensional synthetic functions and Mujoco tasks. We also conduct comprehensive study on different MCMC transition numbers and transition noises to fully examine the performance and robustness of MCMC-BO . We further validate that our proposed method is an effective GP-based BO algorithm as other Multi-Armed Bandit algorithms.

We compare MCMC-BO to state-of-the-art baselines of high-dimensional BO algorithms (TuRBO, LA-MCTS and HesBO) and EA algorithms (CMA-ES and Shiwa). For all BO algorithms, we use Thompson sampling to sample batches in each iteration and discretize the continuous search domain using scrambled Sobolev sequence. To evaluate performance of MCMC-BO, we used TuRBO and LA-MCTS as its BO component. Performance figures show the mean performance of algorithms with one standard error.

High-Dimensional Synthetic Functions

We choose to optimize two popular synthetic problems: Ackley function over domain $[-5,10]^d$, and Rastrigin function over domain $[-5,5]^d$. For dimension d of all functions, we choose $d = 200, 400$ and 800 to evaluate performance over high-dimensional problems, and set same transition number as function dimension. All problems start with 200 initial points and sample a batch of 100 points in each iteration.

Fig 3 suggests that MCMC-BO consistently outperforms other baselines on all functions. In higher dimensions $d = 800$, uniform discretization can not support good exploitation in such dimension, and TuRBO and LA-MCTS degenerate to same level performance as EA algorithms. MCMC-BO achieves the best performance in all selected dimensions by allocating limited action points to more promising region. Note that on Ackley-200d task, Eriksson et al. reports a mean performance of 5 by TuRBO-1 after 10k evaluations(Eriksson et al. 2019), while MCMC-BO with TuRBO-1 reaches a mean performance of 4 after only 3k evaluations under same initial number and batch size.

Mujoco Locomotion Task

Mujoco locomotion tasks are popular benchmarks for reinforcement learning (RL) algorithms(Todorov, Erez, and Tassa 2012). Here we choose Hopper task and Half-Cheetah task in spaces of dimension 33 and 102. To evaluate performance of these sampling-based algorithms, we optimize a linear policy: $\mathbf{a} = \mathbf{W}\mathbf{s}$(Mania, Guy, and Recht 2018). The elements of parameter matrix $\mathbf{W}$ are continuous and in the range of $[-1,1]$. For each proposed policy, the reward is computed over 10 episodes. Both tasks start with 200 initial points and sample a batch of 50 points in each iteration. We set the transition number to 200 on both tasks.

Fig 4 shows the optimization performance of all algorithms. In Hopper task, MCMC-BOwith TuRBO-20 algorithm converges faster than original TuRBO-20. In higher dimensional Half-Cheetah task where EA algorithms fail to find good rewards, MCMC-BO with TuRBO-1 still outperforms other baselines.

Function	MCMC transition noise				
	0.0001	0.0005	0.001	0.005	0.01
Ackley-20d	$-0.51_{\pm 0.08}$	$-0.25_{\pm 0.27}$	$-0.18_{\pm 0.02}$	$-0.17_{\pm 0.02}$	$-0.22_{\pm 0.06}$
Rastrigin-20d	$-5.41_{\pm 2.30}$	$-4.59_{\pm 2.05}$	$-2.13_{\pm 2.01}$	$-2.49_{\pm 0.71}$	$-4.17_{\pm 1.29}$

Table 1: Ablation studies on hyper-parameters of MCMC-BO. The results are the means and standard deviations of the five replicate independent experiments on Ackley-20d and Rastrigin-20d when the transition number is 300 (The result of the Rastrigin function is added by 180).

Ablations on Hyperparameters

There are only two hyperparameters in MCMC-BO: MCMC transition number and noise. Here we conduct ablation study on Ackley-20d function and Rastrigin-20d function. Table 1 shows that the performance of MCMC-BO is not sensitive to transition noise when the transition number is large enough. As transition number increases, the performance steadily improves until convergence at a certain number (see Appendix).

Performance on low-dimensional problems

We compare MCMC-BO with GP-TS, and GP-UCB on 1d and 2d synthetic functions. Fig 5 shows that our proposed method achieves competitive performance against two popular BO algorithms, serving as an effective method in low-dimensional cases.

To a certain extent, MCMC-BO mainly improves the performance by choosing more candidate points on GP obeying a Thompson sampling manner. Compared to directly introducing more points, our proposed method frees the need of

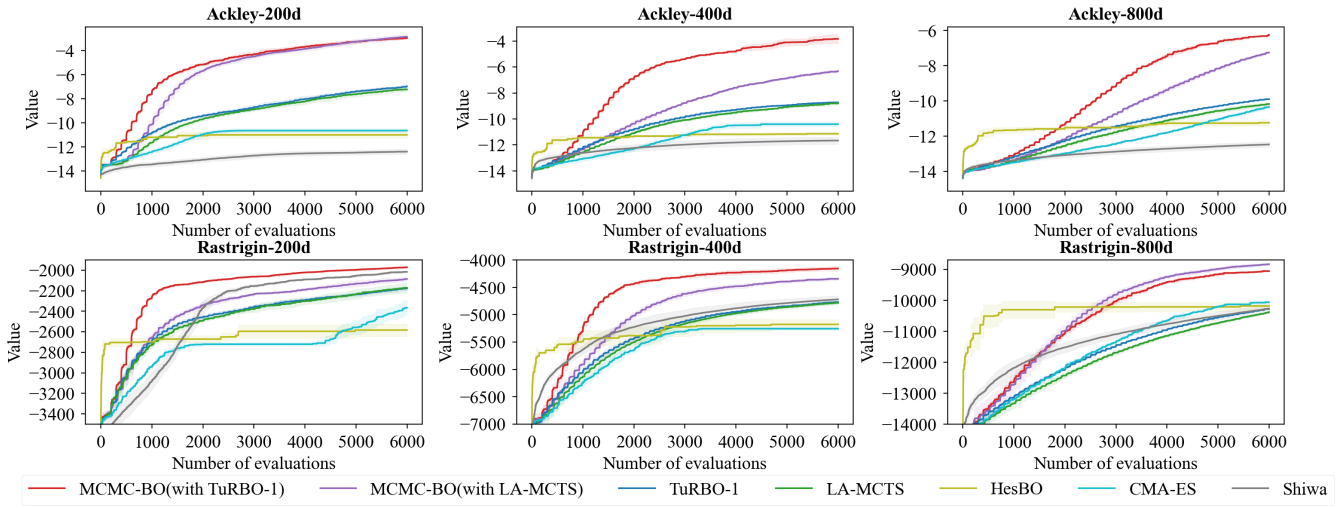


Figure 3: Synthetic functions.

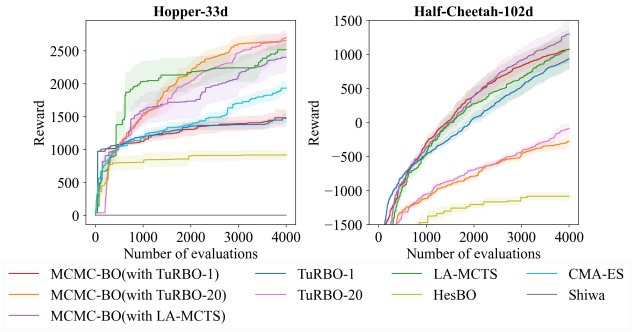


Figure 4: Mujoco locomotion tasks.

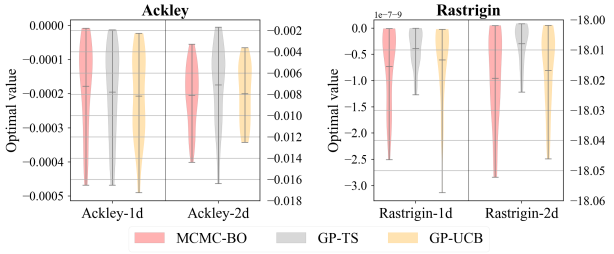


Figure 5: We compare MCMC-BO to GP-TS and GP-UCB on low-dimensional functions. We do 50 replicates of independent experiments. For each method we depict the distribution over the optimal value encoded as a violin plot, with horizontal bars 20% quantiles.

inverting huge matrix, leading to less memory and computation cost. Compared to UCB based acquisition, MCMC-BO is a probability distribution over the domain, more straightforward to perform a batch solution. Meanwhile, GP-UCB only utilize the diagonal information of the covariance matrix of $f(x_i)$ and Thompson sampling use the whole matrix's information at a time. As a compromise, MCMC-BO use a 2×2 sub-matrix at a time in a divide-and-conquer strategy to reduce computation cost and reach an approximation stationary distribution of original Thompson sampling.

Conclusion and Future Work

Sequential optimization in high dimensional space is a task with profound impact in numerical analysis and optimization research. In this paper, we propose MCMC-BO to improve sample efficiency of high dimensional Bayesian optimization. MCMC-BO allows for versatile transitions to promising regions instead of maintaining a huge candidate set, reaching better likelihood of the stationary distribution of the approximated Thompson sampling. Our algorithm, which can be viewed as a GP-based bandit algorithm, yields an effective distribution approximation of Thompson sampling for optimization without the need to compute inverse matrices of thousands of dimensions. We derive the regret bound of MCMC-BO under high-dimensional cases without memory overuse. We also conduct comprehensive evaluations to show that MCMC-BO can improve on existing popular high-dimensional BO baselines.

Future directions include developing parallel-computation mechanisms to improve the computation speed, and constructing Markov chains which can incorporate the used of gradient information from GP to accelerate the convergence rate. Notice that unlike other discretization methods, MCMC-BO doesn't impose any regularity assumptions on the sampling candidate regions. This could be an advantage when combining MCMC-BO with more complex space partition algorithms.

References

Andrieu, C.; De Freitas, N.; Doucet, A.; and Jordan, M. I. 2003. An introduction to MCMC for machine learning. *Machine learning*, 50(1): 5–43.

Bergstra, J.; Bardenet, R.; Bengio, Y.; and Kégl, B. 2011. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

Chowdhury, S. R.; and Gopalan, A. 2017. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, 844–853. PMLR.

Eriksson, D.; and Jankowiak, M. 2021. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, 493–503. PMLR.

Eriksson, D.; Pearce, M.; Gardner, J.; Turner, R. D.; and Poloczek, M. 2019. Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32.

Falkner, S.; Klein, A.; and Hutter, F. 2018. BOHB: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, 1437–1446. PMLR.

Frazier, P. I. 2018. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*.

Gardner, J.; Guo, C.; Weinberger, K.; Garnett, R.; and Grosse, R. 2017. Discovering and exploiting additive structure for Bayesian optimization. In *Artificial Intelligence and Statistics*, 1311–1319. PMLR.

Hansen, N.; Müller, S. D.; and Koumoutsakos, P. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation*, 11(1): 1–18.

Hensman, J.; Fusi, N.; and Lawrence, N. D. 2013. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.

Hernández-Lobato, J. M.; Requeima, J.; Pyzer-Knapp, E. O.; and Aspuru-Guzik, A. 2017. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International conference on machine learning*, 1470–1479. PMLR.

Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2011. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, 507–523. Springer.

Jin, Y.; and Branke, J. 2005. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on evolutionary computation*, 9(3): 303–317.

Kandasamy, K.; Schneider, J.; and Póczos, B. 2015. High dimensional Bayesian optimisation and bandits via additive models. In *International conference on machine learning*, 295–304. PMLR.

Kirschner, J.; Mutny, M.; Hiller, N.; Ischebeck, R.; and Krause, A. 2019. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In *International Conference on Machine Learning*, 3429–3438. PMLR.

Letham, B.; Calandra, R.; Rai, A.; and Bakshy, E. 2020. Re-examining linear embeddings for high-dimensional Bayesian optimization. *Advances in neural information processing systems*, 33: 1546–1558.

Li, C.; Gupta, S.; Rana, S.; Nguyen, V.; Venkatesh, S.; and Shilton, A. 2018. High dimensional Bayesian optimization using dropout. *arXiv preprint arXiv:1802.05400*.

Liu, J.; Moreau, A.; Preuss, M.; Rapin, J.; Roziere, B.; Teytaud, F.; and Teytaud, O. 2020. Versatile black-box optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 620–628.

Lu, X.; Gonzalez, J.; Dai, Z.; and Lawrence, N. D. 2018. Structured variationally auto-encoded optimization. In *International conference on machine learning*, 3267–3275. PMLR.

Mania, H.; Guy, A.; and Recht, B. 2018. Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*.

Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; and Teller, E. 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6): 1087–1092.

Mutny, M.; and Krause, A. 2018. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. *Advances in Neural Information Processing Systems*, 31.

Nayebi, A.; Munteanu, A.; and Poloczek, M. 2019. A framework for Bayesian optimization in embedded subspaces. In *International Conference on Machine Learning*, 4752–4761. PMLR.

Oh, C.; Gavves, E.; and Welling, M. 2018. BOCK: Bayesian optimization with cylindrical kernels. In *International Conference on Machine Learning*, 3868–3877. PMLR.

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

Seeger, M. W.; Williams, C. K.; and Lawrence, N. D. 2003. Fast forward selection to speed up sparse Gaussian process regression. In *International Workshop on Artificial Intelligence and Statistics*, 254–261. PMLR.

Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and De Freitas, N. 2015. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175.

Snelson, E.; and Ghahramani, Z. 2005. Sparse Gaussian processes using pseudo-inputs. *Advances in neural information processing systems*, 18.

Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Snoek, J.; Rippel, O.; Swersky, K.; Kiros, R.; Satish, N.; Sundaram, N.; Patwary, M.; Prabhat, M.; and Adams, R. 2015. Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, 2171–2180. PMLR.

Sobol', I. M. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4): 784–802.

Springenberg, J. T.; Klein, A.; Falkner, S.; and Hutter, F. 2016. Bayesian optimization with robust Bayesian neural networks. *Advances in neural information processing systems*, 29.

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. 2009. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*.

Stanton, S.; Maddox, W.; Delbridge, I.; and Gordon Wilson, A. 2021. Kernel Interpolation for Scalable Online Gaussian Processes. In Banerjee, A.; and Fukumizu, K., eds., *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, 3133–3141. PMLR.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4): 285–294.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033. IEEE.

Tripp, A.; Daxberger, E.; and Hernández-Lobato, J. M. 2020. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33: 11259–11272.

Wang, L.; Fonseca, R.; and Tian, Y. 2020. Learning search space partition for black-box optimization using monte carlo tree search. *Advances in Neural Information Processing Systems*, 33: 19511–19522.

Wang, Z.; Gehring, C.; Kohli, P.; and Jegelka, S. 2018. Batched large-scale Bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, 745–754. PMLR.

Wang, Z.; Hutter, F.; Zoghi, M.; Matheson, D.; and De Feitas, N. 2016. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55: 361–387.

Appendix A Theoretical Analysis

We impose smoothness assumptions on f, the objective function, to begin with our analysis. In the Reproducing Kernel Hilbert Space (RKHS) corresponding to our kernel k, we assume that the unknown $f \in H_k(D)$ satisfies $||f||_k \leq B$. Here $H_k(D)$ is the corresponding RKHS, note that the RKHS norm satisfies $f \in H_k(D)$if and only if $||f||_k < \infty$. Moreover, f should be Lipschitz continuous with constant $B \cdot L$. In our analysis, we work with a discretization D_t of time-varying size as a necessary tool for bounding the regret. However, such discretization is only an analytical tool rather than a practical requirement. we choose a compact and convex domain $D \subset [0, r]^d$ and discretization D_t with size $|D_t| = \left(BLrdt^2\right)^d$ such that $\|x - [x]_t\|_1 \leq rd/BLrdt^2 = 1/BLt^2$ for all $x \in D$, where $L = \sup_{x \in D} \sup_{j \in [d]} \left(\frac{\partial^2 k(p,q)}{\partial p_j \partial q_j}\Big|_{p=q=x} \right)^{1/2}$. This implies, for every $x \in D$, $|f(x) - f([x]_t)| \leq \|f\|_k L \|x - [x]_t\|_1 \leq 1/t^2$. Here we first repeat Definition 1-3 from the main paper.

Definition 1. *From measure theory we recall the concept of a kernel. Consider two measure spaces*$(\Omega, \mathcal{A})$ *and* $(\Omega', \mathcal{A}')$*. The function*

$$K : \Omega \times \mathcal{A}' \quad \rightarrow \quad [0, \infty) \tag{1}$$

is a kernel from $(\Omega, \mathcal{A})$ *to* $(\Omega', \mathcal{A}')$ *if*

- $\omega \mapsto K(\omega, A')$ *is* $\mathcal{A}$*-measurable* $\quad \forall \quad A' \in \mathcal{A}'$
- $A' \mapsto K(\omega, A')$ *is a measure on* $\mathcal{A}' \quad \forall \quad \omega \in \Omega$.

If $K(\omega, \Omega') = 1$*, the kernel is a so-called Markov Kernel.*

Definition 2. *Define* $E^f(t)$ *as the event that for all* $x \in D$*,*

$$|\mu_{t-1}(x) - f(x)| \leq v_t \sigma_{t-1}(x),$$

where $\tilde{c}_t = \sqrt{4 \ln t + 2d \ln\left(BLrdt^2\right)}$ *and* $c_t = v_t\left(1 + \tilde{c}_t\right)$

Definition 3. *Define the set of points with better estimated mean in discretization* D_t *at round* t *as*

$$G_t := \{x \in D_t : \mu_{t-1}(x) \geq f\left([x^\star]_t\right) - v_t \tilde{c}_t \sigma_{t-1}(x)\}$$

This bound measures the difference between function values at the closet point to $x^\star$ *in* D_t *and at the* $t-1$*'s round GP's mean value of x.*

Definition 4. *1. A Markov chain with* n*-step transition probability* $P^n(x, A) = P(\Phi_n \in A \mid \Phi_0 = x)$ *is* φ***-irreducible*** *if there exists a measure* φ *such that*

$$\varphi(A) > 0 \Rightarrow \sum_n P^n(x, A) > 0, \quad \textit{for } x \in \mathrm{X}.$$

2. For a φ*-irreducible chain, any set* A *with* $\varphi(A) > 0$ *contains a* ***small set*** C *such that for* $\delta > 0,\ n > 0$ *and*

probability measure ν concentrated on C,

$$P^n(x,\cdot) \geq \delta\nu(\cdot), \quad \textit{for } x \in C. \tag{2}$$

3. *The chain is **aperiodic** if for some small set C with $\varphi(C) > 0$, the greatest common divisor of all the n such that* (2) *is true is* 1.

4. *The chain Φ with transition probability $P^n(x,\cdot)$ and an invariant distribution ϕ for Φ is **uniformly ergodic** if there exists a sequence $\tau(n)$ converging to 0 such that for all x, $|| P^n(x,\cdot) - \pi ||\leq \tau(n)$.*

Lemma 3. *[7] For any Markov chain the following are equivalent:*

1. *The chain is aperiodic and the following condition hols: there is a probability measure ϕ on $\mathcal{B}(X)$ and $\epsilon > 1,\ \delta > 0,\ m \in \mathbb{Z}_{>0}$ such that when $\phi(A) > \epsilon$, $\inf_{x\in X} P^m(x,A) > \delta$.*

2. *For some small set C we have $\sup_{x\in X} E_x[\tau_C] < \infty$, where τ_C is the first return time to C.*

3. *Φ is uniformly ergodic.*

When any of the above criterion is satisfied, for any x we can bound the rate of convergence by

$$\|P^n(x,\cdot) - \pi\| \leq (1-\delta)^{[n/m]}.$$

Lemma 4. *A chain Φ is π-irreducible if $\pi(y) > 0 \Rightarrow q(x,y) > 0$, for $x \in$ X. The* MCMC-BO*'s chain Φ is aperiodic and every compact set C with $\mu(C) > 0$ is small if $\pi(x)$ and $q(y,x)$ are positive and continuous for all x, y.*

Proof. See [7] □

Lemma 5. *The* MCMC-BO*'s Markov chain is **aperiodic**, **acyle** and **p-irreducible**, it has stationary distribution. And any state x has a finite expectation of first return time, i.e. $E_x[\tau_x] < \infty$ and $E_x[\tau_C] < \infty$.*

Proof. The MCMC-BO's chain Φ has finitely many states and any state x. The final transition matrix denote as $\Phi = Q \cdot \boldsymbol{\alpha}$. For a proposed chain (Brownian motion or uniform distribution) $Q_{i,j} \geq 0,\ \forall\, i,j$. As $\alpha_{i,j} = \min(1, \frac{\Psi(\cdot)}{1-\Psi(\cdot)}) \geq 0 \forall i \neq j$. We denote the transition probability from point i to j as $\Phi_{ij} = \alpha_{ij}Q_{ij} \geq 0 \forall i \neq j$ and $\Phi_{ii} \geq Q_{ii} \geq 0$. Therefore, the chain is acyclic and irreducible. For a finite state chain, it has finite expectation of first return time $E_x[\tau_x] < \infty$ and $E_x[\tau_C] < \infty$. □

Theorem 1. *For the proposed approximated transition method, it is not a reversible Markov chain, but it still has a stationary distribution $\pi(x)$.*

Proof. Intuitively, MH algorithm leverage the ratio of stationary distribution. And as we shown above, for a symmetric proposal chain $q(x_o, x_p) = q(x_p, x_o) = Q(x_o - x_p)$ where $x_o, x_p \in D_t$ whether the ratio $\frac{P(f_t(x_p)\geq f_t(x_o))q(x_p|x_o)}{P(f_t(x_o)\geq f_t(x_p))q(x_o|x_p)}$ is greater than one is only determined by $\mu(x_o) - \mu(x_p)$. Therefore, we can rank the distribution according to $\mu(x)$

given by posterior. For any three point x_i, x_j, x_j satisfying $\mu(x_i) > \mu(x_j) > \mu(x_k)$ if the MH chain is reversible, then $\pi(x_i)P_{ij} = \pi(x_j)P_{ji}$ and should satisfy that

$$\frac{\pi(x_i)}{\pi(x_j)}\frac{\pi(x_j)}{\pi(x_k)}\frac{\pi(x_k)}{\pi(x_i)} = \frac{P_{ij}}{P_{ji}}\frac{P_{kj}}{P_{jk}}\frac{P_{ki}}{P_{ik}} = 1$$

From the given rank $P(f_t(x_i) \geq f_t(x_j)), P(f_t(x_j) \geq f_t(x_k)), P(f_t(x_i) \geq f_t(x_k))$ are greater than $\frac{1}{2}$. Therefore, $\frac{P_{ij}}{P_{ji}}\frac{P_{kj}}{P_{jk}}\frac{P_{ki}}{P_{ik}} = \frac{P(f_t(x_j>x_i))}{P(f_t(x_i>x_j))}\frac{P(f_t(x_k>x_j))}{P(f_t(x_j>x_k))} \cdot 1 < 1$. Contradicts with the assumption. But as discretization set D_t, it has finite state number N_t. From Lemma 5,4, it has a stationary distribution. □

Lemma 1. *For any stationary distribution from the* MCMC-BO*'s chain* $\pi(x)$,$\exists n_\pi, \forall n \geq n_\pi$ *we can reach a close distribution* $\|P^n(x,\cdot) - \pi\| \leq \epsilon$

Proof. This follows from Theorem1 and Lemma3,5 □

Lemma 6. *Let* $\{x_t\}_{t=1}^{\infty}$ *be an* $\mathbb{R}^d$*-valued discrete time stochastic process predictable with respect to the filtration* $\{\mathcal{F}_t\}_{t=0}^{\infty}$*, i.e.,* x_t *is* $\mathcal{F}_{t-1}$*-measurable* $\forall t \geq 1$*. Let* $\{\varepsilon_t\}_{t=1}^{\infty}$ *be a real-valued stochastic process such that for some* $R \geq 0$ *and for all* $t \geq 1$*,* ε_t *is (a)* $\mathcal{F}_t$*-measurable, and (b)* R*-sub-Gaussian conditionally on* $\mathcal{F}_{t-1}$*. Let* $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ *be a symmetric, positive-semidefinite kernel with associated feature map* $\varphi : \mathbb{R}^d \to H_k$ *and RKHS* H_k*. Denote* K_t *as the* $t \times t$ *matrix* $K_t(i,j) = k(x_i, x_j), 1 \leq i, j \leq t$*.*

Let $0 < \delta \leq 1$*. For a given* $\eta > 0$*, with probability at least* $1 - \delta$*, the following holds:* $\forall\ t > 1$*,*

$$\|\varepsilon_{1:t}\|^2_{\left((K_t+\eta I)^{-1}+I\right)^{-1}} \leq 2R^2 \ln \frac{\sqrt{\det\left((1+\eta)I + K_t\right)}}{\delta}.$$

Proof. See [2]. □

Lemma 7. *Letting* $S_t = \sum_{s=1}^{t} \varepsilon_s \varphi(x_s)$ *and define the (possibly infinite dimensional) matrix* $V_t = I + \sum_{s=1}^{t} \varphi(x_s)\varphi(x_s)^T$*, we have, whenever* K_t *is positive definite, that*

$$\|\varepsilon_{1:t}\|_{\left(K_t^{-1}+I\right)^{-1}} = \|S_t\|_{V_t^{-1}},$$

where $\|S_t\|_{V_t^{-1}} := \left\|V_t^{-1/2} S_t\right\|_{H_k}$ *denotes the norm of* $V_t^{-1/2} S_t$ *in the RKHS* H_k*.*

(Note that for any $x \in \mathbb{R}^t$ *and* $A \in \mathbb{R}^{t\times t}$*,* $\|x\|_A := \sqrt{x^T A x}$*. Moreover, if* K_t *is positive definite* $\forall t \geq 1$ *with probability 1, then the conclusion above holds with* $\eta = 0$*).*

Proof. See [2]. □

Lemma 8. *Following the analysis of GP-UCB, we make use of* γ_t*, the maximum information gain at time* t*, to quantitatively bound the variance term* $\sigma_t(x)$*. It is defined as*

$$\gamma_t := \max_{A \subset \Omega:\ |A| = t} I(y_A; f_A),$$

where $I(y_A; f_A)$ denotes the mutual information between $f_A = [f(x)]_{x\in A}$ and $y_A = f_A + \epsilon_A$ such that $\epsilon_A \sim \mathcal{N}(0, \lambda v^2 I)$. $I(y_A; f_A)$ is a function relating to uncertainty reduction after observing y_A from the actions in A. For a compact subset $\Omega \subseteq \mathbb{R}^d$,

$$\gamma_T \in \begin{cases} O\left((\ln T)^{d+1}\right), & \text{if } k_{SE}(x,x') = \exp\left(-\left(2l^2\right)^{-1}\|x-x'\|^2\right). \\ O\left(T^{d(d+1)/(2\nu+d(d+1))}\ln T\right), & \text{if } k_{Matern}(x,x') = \left(2^{1-\nu}/\Gamma(\nu)\right) r^\nu B_\nu(r). \end{cases}$$

where l is a lengthscale parameter, ν controls the smoothness of sample paths, B_ν a modified Bessel function, and $r = (\sqrt{2\nu}/l)\|x-x'\|$ [9].

For every $t \geq 0$, the maximum information gain γ_t, for the points chosen by Algorithm 1 and 2 satisfy, almost surely, the following :

$$\gamma_t \geq \frac{1}{2}\ln\left(\det\left(I + \lambda^{-1}K_t\right)\right)$$
$$\gamma_t \geq \frac{1}{2}\sum_{s=1}^{t}\ln\left(1+\lambda^{-1}\sigma^2_{s-1}(x_s)\right).$$

Lemma 9. *Let $x_1, \dots x_t$ be the points selected by the algorithms. The sum of predictive standard deviation at those points can be expressed in terms of the maximum information gain. More precisely,*

$$\sum_{t=1}^{T}\sigma_{t-1}(x_t) \leq \sqrt{4(T+2)\gamma_T}.$$

Proof. See [2]. □

Lemma 10. *For a Gaussian random variable X with mean μ and standard deviation σ, for any $\beta > 0$, from gaussian tail bound $\mathbb{P}[(X-\mu) \geq t] = \mathbb{P}\left[e^{\lambda(X-\mu)} \geq e^{\lambda t}\right] \leq \frac{\mathbb{E}\left[e^{\lambda(X-\mu)}\right]}{e^{\lambda t}}$, we have that*

$$\mathbb{P}\left[\frac{X-\mu}{\sigma} > \beta\right] \geq \frac{e^{-\beta^2}}{4\sqrt{\pi}\beta} \tag{3}$$

$$\mathbb{P}[X \geq \mu + \beta] \leq e^{-\frac{\beta^2}{2\sigma^2}},\ \forall t \geq 0$$

Definition 5. *Define the set of saturated points S_t in discretization D_t at round t as*

$$S_t := \{x \in D_t : \Delta_t(x) > c_t\sigma_{t-1}(x)\}$$

where $\Delta_t(x) := f\left([x^\star]_t\right) - f(x)$, the difference between function values at the closest point to $x^\star$ in D_t and at x. Clearly $\Delta_t\left([x^\star]_t\right) = 0$ for all t, and hence $[x^\star]_t \in D_t$ is unsaturated at every t.

Lemma 11. *[2] Under the same hypotheses as those of Theorem 6 , let $D \subset \mathbb{R}^d$, and $f : D \to \mathbb{R}$ be a member of the RKHS of real-valued functions on D with kernel k, with RKHS norm bounded by B. Then, with probability at least $1-\delta$, the following holds for all $x \in D$ and $t \geq 1$: $|\mu_{t-1}(x) - f(x)| \leq \left(B + R\sqrt{2\left(\gamma_{t-1} + 1 + \ln(1/\delta)\right)}\right)\sigma_{t-1}(x)$, where γ_{t-1} is the maximum information gain after $t-1$ rounds and $\mu_{t-1}(x), \sigma^2_{t-1}(x)$ are mean and variance of*

posterior distribution defined as in Equation 2, 3, 4, with λ set to $1+\eta$ and $\eta = 2/T$.

Lemma 12. *Consider the stationary distribution as π and the proposal distribution as Q, divide the discretization set D_t into three subgroups D_t^1, D_t^2, D_t^3 which satisfy that for any $x_{D_1} \in D_t^1, x_{D_2} \in D_t^2, x_{D_3} \in D_t^3$,if we have $\mu_{t-1}(x_{D_1}) \geq \mu_{t-1}(x_{D_2}) \geq \mu_{t-1}(x_{D_3})$. We denote $\pi_i = \sum_{x \in D_i} P(x)$. And for the proposal distribution Q, we have that*

$$\pi_3 \leq \frac{Q_{13}\alpha_{13}}{1 - Q_{33}} \tag{4}$$

Proof. Suppose that $\pi = [\pi_1, \pi_2, \pi_3]$ is the stationary distribution of the three subgroups, then we must have that

$$\begin{pmatrix} \pi_1 & \pi_2 & \pi_3 \end{pmatrix} \left(\begin{pmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{pmatrix} \cdot \begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} \\ 1 & 1 & \alpha_{23} \\ 1 & 1 & 1 \end{pmatrix} \right) = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 \end{pmatrix}, \tag{5}$$

where α_{ij} denotes the acceptance probability of transfering group i to group j. By assumption of $\mu_{t-1}(x_{D_i})$, we known that for $i \geq j$, $\alpha_{ij} = 1$. Thus, (5) gives

$$\pi_3 = \pi_1 Q_{13}\alpha_{13} + \pi_2 Q_{23}\alpha_{23} + \pi_3 Q_{33}.$$

Subsituting $\pi_1 = 1 - \pi_2 - \pi_3$, we have

$$\begin{aligned} \pi_3 &= (1 - \pi_2 - \pi_3)Q_{13}\alpha_{13} + \pi_2 Q_{23}\alpha_{23} + \pi_3 Q_{33} \\ \pi_3(1 + Q_{13}\alpha_{13} - Q_{33}) &= Q_{13}\alpha_{13} + \pi_2(Q_{23}\alpha_{23} - Q_{13}\alpha_{13}) \end{aligned}$$

Denote

$$\begin{aligned} (\alpha_{ij})_{\min} &:= \min_{x,x'} \alpha(x, x') \quad \text{s.t. } x \in D_t^i,\ x' \in D_t^j \\ (\alpha_{ij})_{\max} &:= \max_{x,x'} \alpha(x, x') \quad \text{s.t. } x \in D_t^i,\ x' \in D_t^j, \end{aligned}$$

and notice that $\forall\ i, j$, $\pi_2 = 1 - \pi_1 - \pi_3 \leq 1 - \pi_3$ and $\alpha_{ij} \leq 1$. We thus have

$$\begin{aligned} \pi_3(1 + Q_{13}(\alpha_{13})_{\min} - Q_{33}) &\leq Q_{13}(\alpha_{13})_{\max} - \pi_2 Q_{13}(\alpha_{13})_{\min} + \pi_2 Q_{23}\alpha_{23} \\ &\leq Q_{13}(\alpha_{13})_{\max} - \pi_2 Q_{13}(\alpha_{13})_{\min} + (1 - \pi_3)Q_{23} \\ \pi_3(1 + Q_{13}(\alpha_{13})_{\min} - Q_{33} + Q_{23}) &\leq Q_{13}(\alpha_{13})_{\max} - \pi_2 Q_{13}(\alpha_{13})_{\min} + Q_{23} \\ \pi_3 &\leq \frac{Q_{13}(\alpha_{13})_{\max} - \pi_2 Q_{13}(\alpha_{13})_{\min} + Q_{23}}{1 + Q_{13}(\alpha_{13})_{\min} - Q_{33} + Q_{23}} \\ &\leq \frac{Q_1 3(\alpha_{13})_{max}}{1 - Q_{33}} \end{aligned}$$

□

W.L.O.G., suppose that Q, the proposal distribution, is a uniform, then following the assumptions of GP-TS, we

can assume that the size of D_t is approximately $c \cdot T^2$, for c a constant. Denote $|m|$ as the number of elements corresponding to π_1, then we have

$$\pi_3 \leq \frac{Q_{13} \cdot \alpha_{13}}{1 - Q_{33} + Q_{13}\alpha_{13}} \leq \frac{Q_{13}\alpha_{13}}{1 - Q_{33}} = \frac{1 - \frac{m}{Ct^2}}{\frac{m}{Ct^2}} \cdot \alpha_{13}.$$

Lemma 13. *Now we define* $D_t^3 := S_t := \{x \in D_t : \Delta_t(x) > c_t\sigma_{t-1}(x)\}$ *and* $D_t^1 := \{x \in D_t : \Delta_t(x) < v_t\sigma_{t-1}(x)\} \in G_t$. *Then we have*

$$\frac{\mu_3 - \mu_1}{\sigma_1 + \sigma_3} \leq -k_t$$

and $k_t = (\tilde{c}_t - 2)v_t \in O(\sqrt{\ln(BLrdt^2)((\ln T)^{d+1})})$

Proof. Recall the definition of $E^f(t)$ and S_t so that for a point x, we have $|f(x) - \mu_{t-1}(x)| \leq v_t\sigma_{t-1}(x)$ and $f(x) \leq f(x^*) - c_t\sigma_{t-1}(x)$. We define a set W_t such that for $x \in W_t$, $\mu_{t-1}(x) \leq f(x^*) - c_t\sigma_{t-1}(x) + v_t\sigma_{t-1}(x) = f(x^*) - v_t\tilde{c}_t\sigma_{t-1}(x)$. Then, for $x_3 \in W_t$,

$$\mu_3 + k_t\sigma_3 \leq f(x^*) - v_t\tilde{c}_t\sigma_3 + k_t\sigma_3.$$

On the other hand, for a point x_1 satisfying

$$f(x) \geq f(x^*) - v_t\sigma_{t-1}(x),$$

and as have that $x_1 \in G_t \in D_t \backslash S_t$ .Then, by definition of S_t, we have that

$$\begin{aligned} \mu_1 + v_t\sigma_1 &\geq f(x^*) - v_t\sigma_1 \\ \mu_1 &\geq f(x^*) - 2v_t\sigma_1 \\ \Rightarrow \mu_1 - k_t\sigma_1 &\geq f(x^*) - 2v_t\sigma_1 - k_t\sigma_1. \end{aligned}$$

Then we can solve for k_t by aligning the two inequalities

$$\begin{aligned} \mu_t(x) + k_t\sigma_t(x) &\leq \mu_t(x) - k_t\sigma_t(x) \\ \Rightarrow f(x^*) - 2v_t\sigma_t(x) - k_t\sigma_t(x) &\geq f(x^*) - v_t\tilde{c}_t\sigma_t(x) + k_t\sigma_t(x) \\ \Rightarrow -2v_t\sigma_t(x) - k_t\sigma_t(x) &\geq v_t\tilde{c}_t\sigma_t(x) + k_t\sigma_t(x). \end{aligned}$$

Hence, we can of course take $k_t = (\tilde{c}_t - 2)v_t$. Since $\tilde{c}_t = \sqrt{4\ln t + 2d\ln(BLrdt^2)}$ and $v_t = B + R\sqrt{2(\gamma_{t-1} + 1 + \ln(2/\delta))}$, and $\gamma_T \in O\left((\ln T)^{d+1}\right)$, then we obtain the previous conclusion by substituting in $k_t \geq \sqrt{4d\ln(BLrdt^2)\gamma_t} \geq O(\sqrt{\ln(BLrdt^2)((\ln T)^{d+1})})$. □

Theorem 2. *For discretization D_t stationary distribution π and superior set*

$$\Xi_t := \{\mu_{t-1}(x) \geq f(x^\star) - \epsilon\}$$

we have that $\forall \epsilon > 0, \exists\ t$, s.t. $\pi(\Xi_t) > q$ where q is a constant from the transition matrix.

Proof. This follows from lemma 13 and continuity. □

Lemma 2. *For any history $\mathcal{H}'_{t-1}$ such that $E^f(t)$ is true,*

$$\mathbb{P}\left[\Xi_t \in G_t\right] \geq 1 - 1/t^2.$$

Theorem 3. *For any filtration $\mathcal{F}'_{t-1}$ such that $E^f(t)$ is true, $\mathbb{P}\left[x_t \in G_t \mid \mathcal{F}'_{t-1}\right] \geq p - 2/t^2$*

Proof. From TS and the posterior probability given by GP, we denote μ_i, σ_i corresponding to $\mu_{t-1}(x_i)$ and $\sigma_{t-1}(x_i)$ for ease of notation. The acceptance probability is given by $\alpha_{13} = \frac{\Phi(\frac{\mu_3-\mu_1}{\sqrt{\sigma_1^2+\sigma_3^2-2\rho\sigma_1\sigma_3}})}{1-\Phi(\frac{\mu_3-\mu_1}{\sqrt{\sigma_1^2+\sigma_3^2-2\rho\sigma_1\sigma_3}})}$ (here ρ denotes the correlation coefficient).

Suppose we could obtain a bound on $O(\gamma_T)$, i.e. we proceed by assuming $O(\gamma_T) \leq k_t$. Notice that since $\Phi(\frac{\mu_3-\mu_1}{\sqrt{\sigma_1^2+\sigma_3^2-2\rho\sigma_1\sigma_3}}) \leq \Phi(\frac{\mu_3-\mu_1}{\sigma_1+\sigma_3})$, and by $\Phi(x) \leq \exp(-x^2/2)$, we have

$$\alpha_{13} \leq \frac{e^{\frac{\mu_3-\mu_1}{\sigma_1+\sigma_3}}}{1 - e^{\frac{\mu_3-\mu_1}{\sigma_1+\sigma_3}}}.$$

then we can bound π_3 by

$$\begin{aligned}\pi_3 &\leq \frac{ct^2 - m}{m} \cdot \frac{e^{-k_t^2/2}}{1 - e^{-k_t^2/2}} \\ &= \frac{ct^2 - m}{m} \frac{1}{e^{k_t^2/2} - 1},\end{aligned}$$

and

$$\lim_{t\to\infty} \frac{ct^2 - m}{m} \frac{1}{e^{k_t^2/2} - 1} \in o(1)$$

We then verify that our previous assumptions regarding $\frac{\mu_3-\mu_1}{\sigma_1+\sigma_2} \leq -k_t$ is correct, which is equivalent to verifying that

$$\frac{\mu_3 - \mu_1}{\sigma_1 + \sigma_3} \leq -k_t \iff \mu_3 + k_t\sigma_3 \leq \mu_1 - k_t\sigma_1.$$

$$\begin{aligned}\lim_{t\to\infty} \frac{Q_{13}\alpha_{13}}{1 - Q_{33}} &\leq \lim_{t\to\infty} (BLr)^d t^{2d} \frac{1}{e^{k_t^2/2} - 1} \\ &\leq \frac{(BLr)^d t^{2d}}{e^{\ln(BLrdt^2)(\ln T)^{d+1}}} \\ &= O\left(e^{\left(2dln(t) - \ln\left(BLrdt^2\right)(\ln T)^{d+1}\right)}\right) = o(1)\end{aligned} \tag{6}$$

And therefore we can bound π_3 successfully and $\mathbb{P}\left[x_t \in G_t \mid \mathcal{F}'_{t-1}\right]$ can then be bounded. □

Theorem 4. *Let $\delta \in (0,1)$, $D \subset [0,r]^d$ be compact and convex, $\|f\|_k \leq B$ and $\{\varepsilon_t\}_t$ a conditionally R-sub-Gaussian sequence. Running GP-TS for a function f lying in the RKHS $H_k(D)$ and with decision sets D_t chosen as above, with probability at least $1-\delta$, the regret of* MCMC-BO *satisfies* $\tilde{O}\left(\gamma_T\sqrt{dT}\right)$

Proof. From here the proof follows the one in [2] from Lemma 9-13. □

Appendix B Combining MCMC-BO with Existing BO Algorithms

In Algorithm 2 we present MCMC-BO combined with any other existing BO algorithms over continuous domain. For various BO algorithms, the main difference can reduce to their explicit or implicit acquisition functions. MCMC-BO is able to take points proposed by BO methods in the MCMC transition to achieve more promising region. Popular BO acquisition functions also provide good initial positions, reducing the number of MCMC transitions.

Algorithm 2 MCMC-BO with existing BO algorithms

Input Initial dataset $\mathcal{D}_0$, Acquisition function A, batch size m, MCMC transition number N, transition noise $\boldsymbol{\sigma}$
  for $t = 0, 1, \cdots$ **do**
    Update posterior distribution on f using $\mathcal{D}_t$
    Create descretized candidate points $\boldsymbol{x}_{\text{cand}}$ from continuous search domain
    Propose a batch $\boldsymbol{x}_t^0 = \text{argmax}_{\boldsymbol{x}_{\text{cand}}} A(\boldsymbol{x}_{\text{cand}}|\mathcal{D}_t)$
    {MCMC transition start}
    for $i = 0$ to $N-1$ **do**
      for $k = 0$ to $m-1$ **do**
        Sample $u \sim \text{Unif}[0,1]$
        Sample $x_{tk}^{i+1} \sim x_{tk}^i + \mathcal{N}(0, \boldsymbol{\sigma})$
        if $u \geq \min\left\{1, \frac{P(f_t(x_{tk}^{i+1}) \geq P(f_t(x_{tk}^i))q(x_{tk}^{i+1}|x_{tk}^i)}{P(f_t(x_{tk}^i) \geq P(f_t(x_{tk}^{i+1}))q(x_{tk}^i|x_{tk}^{i+1})}\right\}$ **then**
          $x_{tk}^{i+1} \leftarrow x_{tk}^i$ // Reject the transition
        end if
      end for
    end for
    {MCMC transition end}
    Observe $\boldsymbol{y}_t = f(\boldsymbol{x}_t^N)$
    $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup (\boldsymbol{x}_t^N, \boldsymbol{y}_t)$
  end for

Appendix C Experimental Details

C.1 Algorithm Implementation

MCMC-BO: We use Gpytorch to implement GP inference in MCMC-BO[4]. When combining MCMC-BO with TuRBO and LA-MCTS, we set transition noise as $0.008 * \boldsymbol{L}$, where $\boldsymbol{L}$ is the size vector of trust region.

TuRBO: TuRBO is implemented based on tutorials from Botorch[1]. While the tutorial only has incomplete demostration of TuRBO-1, we further modify the original code to TuRBO-M version. The hyperparameters are set as defaut setting in authors' reference implementation[3].

LA-MCTS: LA-MCTS, we refer to authors' reference implementations, and use TuRBO-1 as its local BO solver[10].

HesBO: For HesBO, we refer to authors' reference implementations[8]. We transformed defualt GP component into Gpytorch version for faster inference speed on GPU. We set the embedding dimension to 20 for all tasks

CMA-ES: We use pycma[1] to implement CMA-ES, and use defaut setting except setting population size eqauls to batch size.

Shiwa: We use Nevergrad[2] to implement Shiwa, and use default setting to run experiments.

We run BO methods on one Nvidia A100 GPU for CUDA acceleration. All algorithms get initial N_{init} points from Latin hypercube design[6]. For TuRBO-M, each trust region gets $\frac{N_{\text{init}}}{M}$ initial points, where M is the number of trust regions. In acquisition maximiation of every iteration, we discretize the search domain using 5,000 points from scrambled Sobolev sequence.

C.2 Synthetic Functions

The results on all synthetic functions are computed by 5 repeats. Besides Ackley and Rastrigin problems in main paper, Figure 6 shows the additional experiment on Levy function over domain $[-5, 5]^d$. We evaluate algrithms on $d = 200, 400, 800$. MCMC-BO still outperforms all other baselines in all selected dimensions.

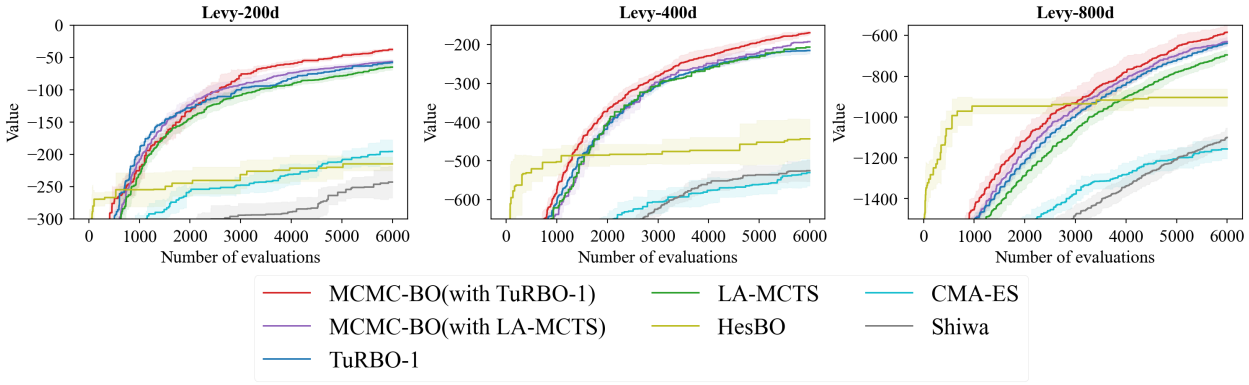


Figure 6: Levy Function.

C.3 Mujoco Locomotion Tasks

We use the mean and standard deviation from [5] to normalize observations of Mujoco task. While obsevations in Mujoco are noisier than synthetic functions, the results on all Mujoco functions are computed by 30 repeats.

C.4 Ablation Study

The results on ablation study are computed by 5 repeats. Table 2 shows the optimization performance under different transition numbers. Within hundreds of transitions, MCMC-BO is able to achieve good performance.

[1]https://github.com/CMA-ES/pycma
[2]https://github.com/facebookresearch/nevergrad

Function	MCMC transition number				
	20	100	300	500	800
Ackley-20d	$-0.38_{\pm 0.21}$	$-0.12_{\pm 0.06}$	$-0.18_{\pm 0.02}$	$-0.26_{\pm 0.23}$	$-0.12_{\pm 0.14}$
Rastrigin-20d	$-5.27_{\pm 2.50}$	$-4.20_{\pm 2.23}$	$-2.13_{\pm 2.01}$	$-3.06_{\pm 2.09}$	$-3.51_{\pm 3.20}$

Table 2: Ablation studies on hyper-parameters of MCMC-BO. The results are the means and standard deviations of the five replicate independent experiments on Ackley-20d and Rastrigin-20d when the transition noise is 0.001 (The result of the Rastrigin function is added by 180).

C.5 Low-dimensional Perfomance

In this section, we compare MCMC-BO diagram against two popular GP-based acquisition functions, GP-UCB and GP-TS, under 1d and 2d Ackley problems. We implement GP-UCB and GP-TS using Gpytorch, and use 5,000 candidate points from scrambled Sobolev sequence to discretize the input space. We record the best points found after 3000 function evaluations. The results on low-dimensional performance are computed by 50 repeats.

C.6 MCMC Transition Demonstration

To better demonstrate the behavior of MCMC in optimization, we visualize the transfer process of 1,000 candidate points in MCMC-BO on Ackley-2d function(see gif file in the supplementary). we use 200 points to initialize MCMC-BO, and sample a batch of 100 points at each iteration. The transition number and transition noise are 2,000 and 0.01. Our MCMC transition demonstration is based on GP posterior after 800 function evaluations.

References

[1] Maximilian Balandat et al. "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization". In: *Advances in Neural Information Processing Systems 33*. 2020. URL: http://arxiv.org/abs/1910.06403.

[2] Sayak Ray Chowdhury and Aditya Gopalan. "On kernelized multi-armed bandits". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 844–853.

[3] David Eriksson et al. "Scalable global optimization via local bayesian optimization". In: *Advances in neural information processing systems* 32 (2019).

[4] Jacob R Gardner et al. "GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration". In: *Advances in Neural Information Processing Systems*. 2018.

[5] Horia Mania, Aurelia Guy, and Benjamin Recht. "Simple random search provides a competitive approach to reinforcement learning". In: *arXiv preprint arXiv:1803.07055* (2018).

[6] Michael D McKay, Richard J Beckman, and William J Conover. "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code". In: *Technometrics* 42.1 (2000), pp. 55–61.

[7] Kerrie L Mengersen and Richard L Tweedie. “Rates of convergence of the Hastings and Metropolis algorithms”. In: *The annals of Statistics* 24.1 (1996), pp. 101–121.

[8] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. “A framework for Bayesian optimization in embedded subspaces”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4752–4761.

[9] Niranjan Srinivas et al. “Gaussian process optimization in the bandit setting: No regret and experimental design”. In: *arXiv preprint arXiv:0912.3995* (2009).

[10] Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. “Learning search space partition for black-box optimization using monte carlo tree search”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 19511–19522.