Learning Region of Interest for Bayesian Optimization with Adaptive Level-Set Estimation

Fengxue Zhang ZHANGFX@UCHICAGO.EDU
Jialin Song JIALINS@NVIDIA.COM
James Bowden JBOWDEN@CALTECH.EDU
Alexander Ladd LADD12@LLNL.GOV
Yisong Yue YYUE@CALTECH.EDU
Thomas A. Desautels DESAUTELS2@LLNL.GOV
Yuxin Chen CHENYUXIN@UCHICAGO.EDU

Abstract

Bayesian optimization (BO) has been applied in black-box optimization tasks which are common in domains including optimal experimental design, self-tuning systems and hyperparameter optimization. Despite the strong theoretical guarantees for BO in the canonical settings, it remains a significant challenge for BO to scale to high-dimensional and non-stationary scenarios. Recent works attempt to exploit the locality of the black-box functions in BO — by partitioning the search space, these algorithms learn a Gaussian process model over regions of interests that better captures the locality of the black-box function. Various heuristics have been proposed along this direction, most relying on additional hyperparameters (e.g., number of local regions/models to be considered, number of examples in each partition, etc.) to be fine-tuned for specific tasks. In this paper, we propose a simple yet effective framework for adaptively learning regions of interest for Bayesian optimization. Our model maintains two Gaussian processes: one global model for identifying the (local) regions of interest as (adaptive) level-sets; the other model for acquiring data in the high-confidence regions of interest. This gives us the benefit of a non-parametric model for learning multiple regions of interest, while having few hyperparameters to maintain. We demonstrate the effectiveness of BALLET empirically on both synthetic and real-world optimization tasks.

Keywords: Bayesian Optimization, Level Set, Partition, Local Optimization

1. Introduction

Bayesian optimization (BO) is a popular statistic-model-based sequential optimization method in various fields of science and engineering, including scientific experimental design (Yang et al., 2019), robotics planning (Berkenkamp et al., 2016; Sui et al., 2018), self-tuning systems (Zhang et al., 2022) and hyperparameter optimization (Snoek et al., 2012). These applications usually involve optimizing a black-box function that is expensive to evaluate, where the statistics-guided efficient optimization algorithm is desired. The common practice in Bayesian optimization is to employ Gaussian processes (GPs) (Rasmussen and Williams, 2006) as a statistic surrogate model for the unknown objective function due to its conjugacy in Bayesian inference and promising capability in terms of learning and inference which allows defining effective acquisition function.

The proper choice of hyperparameters and acquisition function for BO allow theoretical guarantee on its performance under certain smoothness assumptions (Srinivas et al., 2009; Wang and Jegelka, 2017; Wang et al., 2016a). However, the high-dimensional, large-scale, and heterogeneous character of the real-world optimization tasks challenges BO and degrades its performance. Besides the well-known curse of dimensionality (Bengio et al., 2005), the heterogeneity in large-scale tasks challenges the practice of learning a global GP and selecting evaluation candidates with a global acquisition function while lacking training data in the optimization setting (Eriksson et al., 2019). Meanwhile, purely relying on local characteristics in optimization has been proven to be even less effective due to the ignorance of the correlations on observations that are normally captured by the GP when applying an appropriate kernel. The trade-off between capturing the locality and the correlations especially when lacking training data emerges as a critical problem in applying BO in real-world applications where the global smoothness assumption doesn't hold.

Historically, various partitioning-based BO methods have been proposed to tackle this challenge. These methods learn the regions of interest with heuristics that normally introduce additional complexity to the surrogate models, which could incur challenges in fine-tuning the hyperparameters of the heuristics. Some typical examples include the number of the regions of interest (Eriksson et al., 2019), the maximum leaf size in the tree-structured partitioning methods, and the methods to generalize the partition learned on the accumulated observations to the whole search space (Munos, 2014). In contrast to existing work, we consider a non-parametric model for partitioning the search space, which has shown remarkable performance in real-world tasks while having few hyperparameters to maintain. Figure 1 illustrates the optimization framework, with contributions summarized below.

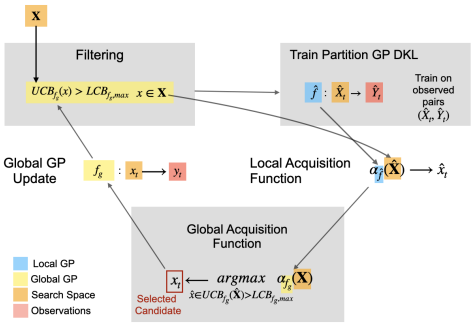


Figure 1: Illustration of the optimization framework. The algorithm filters the search space $\mathbf{X}$ using the UCB and LCB estimated by a global GP. Then it feeds another GP the region of interest of the search space $\hat{\mathbf{X}}$ and the filtered historical observations $(\hat{\mathbf{X}}_t, \hat{\mathbf{Y}}_t)$. Global optimization at each iteration is conducted on the adaptive partition using this GP and its acquisition function.

- We propose a novel algorithm for learning regions of interest for BO. Concretely, we leverage a global estimation of the point-wise upper confidence bound (UCB) and lower confidence bound (LCB) of the black-box function, to adaptively partition the search space for the downstream global optimization task. The partitioning algorithm enables us to identify the local search space that contains the optimum with high probability.
- One can plug in any existing global BO algorithm into our adaptive partitioning framework to tackle high-dimensional large-scale optimization tasks. It offers a statistical guarantee on the distribution of the sub-domains, which leaves room for potential performance analysis.
- We demonstrate the effectiveness of the proposed algorithm with an empirical study on several synthetic and real-world optimization tasks.

2. Related Work

High-dimensional Bayesian optimization. BO often uses Gaussian processes as the function class to parametrize the black-box function because GPs can estimate the uncertainty while fitting the function. However, GPs are difficult to fit in high-dimensional input space so classical BO

algorithms need modifications for high-dimensional functions (Djolonga et al., 2013). To tackle the curse of dimensionality, a class of methods impose additional structures, such as additive GPs, to address the challenge of training a single global GP. LineBO restricts its search space to a one-dimensional subspace with reduced sample complexity at each step (Kirschner et al., 2019). Another idea is to embed the high-dimensional input space into a low-dimensional subspace (Wang et al., 2016b). Our proposed method relates most closely to methods with input space partitions (Eriksson et al., 2019; Wang et al., 2020; Sazanovich et al., 2021). TurBO (Eriksson et al., 2019) maintains a collection of local GPs and allocate queries with a multi-armed bandit procedure. LA-MCTS (Wang et al., 2020) learns a partition of the input space and uses Monte Carlo tree search (MCTS) to decide a subspace to apply BO. Compared with the proposed BALLET, these partitioning methods rely on heuristics and add extra complexity to the optimization task with hyperparameters of these heuristics, e.g., TurBO relies on the number of trust region and LA-MCTS relies on a leaf size, a hyperparameter in UCB for the subspace selection and one for the partitioning algorithm.

Partition-based Bayesian active learning and optimization Partition-based methods are common in BO with safety constraints (Sazanovich et al., 2021; Sui et al., 2018; Makarova et al., 2021). These methods use LCB from GPs to partition the input space into safe and unsafe subspaces. Subsequent optimizaton queries are restricted to the safe subspaces only. Another related work is the level set estimation (LSE) method by Gotovos et al. (2013), where the authors use both UCB and LCB to narrow down regions where a particular function value is likely to exist. Our method inherit the spirit of LSE to leverage the confidence interval to adaptively partition the search space.

Partition-based optimization methods. More broadly speaking, partitioning the input space is a general strategy employed by several optimization methods (Munos, 2011, 2014; Shahriari et al., 2016; Merrill et al., 2021; Kawaguchi et al., 2016). Simultaneous optimistic optimization (SOO) algorithm (Munos, 2011, 2014), which is a non-Bayesian approach that intelligently partitions the space based on observed experiments to effectively balance exploration and exploitation of the objective. A modification of SOO, named Locally Oriented Global Optimization (LOGO) (Kawaguchi et al., 2016), achieves both fast convergence in practice and a finite-time error bound in theory. However, in high-dimensional spaces, these non-Bayesian approaches suffer more compared to Bayesian ones (Merrill et al., 2021).

3. Bayesian Optimization with Adaptive Level-Set Estimation

Bayesian Optimization Formally, the Bayesian optimization algorithm sequentially optimizes a function $f : \mathbf{X} \to \mathbb{R}$, where $\mathbf{X} \subseteq \mathbb{R}^d$ is the search space. At iteration t, the algorithm maintains a Gaussian process $\mathcal{GP}$ as the surrogate model, picks a point $\mathbf{x}_t \in \mathbf{X}$ by maximizing the acquisition function $\alpha : \mathbf{X} \to \mathbb{R}$, and observe the function value perturbed by additive noise: $y_t = f(\mathbf{x}_t) + \epsilon_t$ with $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ being i.i.d. Gaussian noise. The goal is to maximize the sum of rewards $\sum_{t=1}^{T} f(\mathbf{x}_t)$ over T iterations, or equivalently, to minimize the *cumulative regret* $R_T := \sum_{t=1}^{T} r_t$, where $r_t := \max_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) - f(\mathbf{x}_t)$ denotes the *instantaneous regret*. Another common performance metric in BO is the *simple regret* $r_T^* = \max_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x}) - \max_{t \le T} f(\mathbf{x}_t)$.

Global Estimation Existing works use heuristics to partition the historical observations $\mathbf{D}_t = \{X_t, Y_t\}$ first and then generalize it to the whole search space $\mathbf{X}$ (Wang et al., 2020; Eriksson et al., 2019). Here $Y_t = \{y_1, ..., y_t\}$ and $X_t = \{\mathbf{x}_1, ..., \mathbf{x}_t\}$. The heuristics could introduce unnecessary complexity and potentially incur the loss on the accuracy of the partitioning. Instead, we propose to learn the global partitioning on $\mathbf{X}$ with Deep Kernel Learning (DKL) (Wilson et al., 2016) due

to its scalability enabling global estimation of the black-box function. The underlying global function $f_g := f$ is assumed to be drawn from a global Gaussian process $\mathcal{GP}_{f_g}(m_{f_g}(\mathbf{x}), k_{f_g}(\mathbf{x}, \mathbf{x}'))$ parameterized by θ_{f_g}, where $m_{f_g}(\mathbf{x})$ is the mean function and $k_{f_g}(\mathbf{x}, \mathbf{x}')$ is the covariance function. The algorithm learns a latent space mapping $q : \mathbf{X} \to \mathbf{Z}$ on a neural network to convert the input space $\mathbf{X}$ to the latent space $\mathbf{Z}$, and constructs an objective mapping $h : \mathbf{Z} \to \mathbb{R}$ such that $f(\mathbf{x}) \approx h(q(\mathbf{x}))$, $\forall \mathbf{z} \in \mathbf{Z}$. The neural network q and the base kernel k together are regarded as a *deep kernel*, denoted by $k_{f_g}(\mathbf{x}, \mathbf{x}') = k(q(\mathbf{x}), q(\mathbf{x}'))$ (Wilson et al., 2016). The deep kernel is trained by maximizing the negative log-likelihood (NLL) $-\log(P(\mathbf{y}_t|X_t, \theta_{f_g,t})) = -\frac{1}{2}\mathbf{y}_t^\top(\mathbf{K}_{f_g,t} + \sigma^2 I)^{-1}\mathbf{y}_t - \frac{1}{2}\log|(\mathbf{K}_{f_g,t} + \sigma^2 I)| - \frac{t}{2}\log(t)$ which is the learning objective for the kernel (Rasmussen and Williams, 2005). In addition to DKL, we use the unlabeled dataset which is sampled from $\mathbf{X}$ to pre-train an Auto-Encoder and use the parameters of its encoder to initialize the neural network q following the protocol described by Ferreira et al. (2020)

At iteration t, given the selected points $\mathbf{D}_t$, the posterior over f_g also takes the form of a GP, with mean $\mu_{f_g,t}(\mathbf{x}) = k_{f_g,t}(\mathbf{x})^\top(\mathbf{K}_{f_g,t} + \sigma^2 I)^{-1}\mathbf{y}_t$ and covariance $k_{f_g,t}(\mathbf{x}, \mathbf{x}') = k_{f_g}(\mathbf{x}, \mathbf{x}') - k_{f_g,t}(\mathbf{x})^\top(\mathbf{K}_{f_g,t} + \sigma^2 I)^{-1}k_{f_g,t}(\mathbf{x}')$, where $k_{f_g,t}(\mathbf{x}) = [k_{f_g}(\mathbf{x}_1, \mathbf{x}), \ldots, k_{f_g}(\mathbf{x}_t, \mathbf{x})]^\top$ and $\mathbf{K}_{f_g,t} := [k_{f_g}(\mathbf{x}, \mathbf{x}')]_{\mathbf{x},\mathbf{x}' \in \mathbf{D}_t}$ is the positive definite kernel matrix (Rasmussen and Williams, 2005).

Region of Interests Filtering The global $\mathcal{GP}_{f_g}$ enables a filtering on $\mathbf{X}$ to locate the region of interest $\hat{\mathbf{X}}$. It is desired for $\hat{\mathbf{X}}$ that with high probability, $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathbf{X}} f(\mathbf{x})$ is contained by $\hat{\mathbf{X}}$, while $|\hat{\mathbf{X}}| \ll |\mathbf{X}|$. The objective function defined on $\hat{\mathbf{X}}$, which is denoted by $\hat{f}$, is therefore of reduced complexity and easier to be captured by a local Gaussian process $\mathcal{GP}_{\hat{f}}$ during the optimization. Specifically, we leverage the confidence interval of the global $\mathcal{GP}_{f_g}$ to define the upper confidence bound $UCB_{f_g}(\mathbf{x}) = \mu_{f_g,t}(\mathbf{x}) + \beta^{1/2}\sigma_{f_g,t}(\mathbf{x})$ and lower confidence bound $LCB_{f_g}(\mathbf{x}) = \mu_{f_g,t}(\mathbf{x}) - \beta^{1/2}\sigma_{f_g,t}(\mathbf{x})$, where $\sigma_{f_g,t}(\mathbf{x}) = k_{f_g,t}(x, x)$ and β acts as an scaling factor. Then the maximum of global lower confidence bound $LCB_{f_g,max} = \max_{\mathbf{x} \in \mathbf{X}} LCB_{f_g}(\mathbf{x})$ can be used as the threshold, and we attain $\hat{\mathbf{X}} = \{x \in \mathbf{X} |\ UCB_{f_g}(\mathbf{x}) > LCB_{f_g,max}\}$ as the region of interest. The historical observation on this subset is denoted as $\hat{\mathbf{D}} = \{(\mathbf{x}, y) \in \mathbf{D} | \mathbf{x} \in \hat{\mathbf{X}}\}$.

At each iteration, the proposed algorithm Bayesian Optimization with Adaptive Level-Set Estimation (BALLET) conducts the region of interest filtering and then local BO on $\hat{\mathbf{X}}$ using the local Gaussian process $\mathcal{GP}_{\hat{f}}$ and local acquisition function $\alpha_{\hat{f}}$ as the global optimization. Our algorithm is presented in Algorithm 1.

Algorithm 1 **B**ayesian Optimization with **A**daptive **L**eve**l**-Set **E**s**t**imation (BALLET)

1: **Input**:Search space $\mathbf{X}$, initial observation $\mathbf{D}_0$, horizon T;
2: **for** $t = 1$ *to* T **do**
3: Learn the global estimation $\mathcal{GP}_{f_g,t}$: $\theta_{f_g,t} \leftarrow \arg\max_{\theta_{f_g}} -\log(P(\mathbf{y}_t|X_{t-1}, \theta_{f_g}))$
4: Partition by region of interest filtering: $\hat{\mathbf{X}}_t \leftarrow \{\mathbf{x} \in \mathbf{X} |\ UCB_{f_g,t}(\mathbf{x}) > LCB_{f_g,t,max}\}$
5: Partition the historical observation: $\hat{\mathbf{D}} = \{(\mathbf{x}, y) \in \mathbf{D} | \mathbf{x} \in \hat{\mathbf{X}}\}$.
6: Learn the local Gaussian process: $\mathcal{GP}_{\hat{f},t}$: $\theta_{\hat{f},t} \leftarrow \arg\max_{\theta_{\hat{f}}} -\log(P(\mathbf{y}_t|\hat{X}_{t-1}, \theta_{\hat{f}})$
7: Optimize the local acquisition function: $\mathbf{x}_{t+1} \leftarrow \arg\max_{\mathbf{x} \in \hat{\mathbf{X}}} \alpha_{\hat{f}}(\mathbf{x}))$
8: Update $\mathbf{D}$: $\mathbf{D}_{t+1} \leftarrow \mathbf{D}_t \cup \{(\mathbf{x}_{t+1}, y_{t+1})\}$
9: **Output**: $\max_t y_t$

Remark 1 BALLET *is not assuming that the resulting* $\hat{\mathbf{X}}$ *is composed of one single cluster. If there are two or more clusters in* $\hat{\mathbf{X}}$*,* BALLET *learns a single GP and optimizes on all these localities at the same time. Intuitively it aims at conducting (local) BO on the top tier (which could consist of multiple regions) of the unknown function.This mechanism avoids being overconfident to identify only one region of interest or the need to manually specify the number of clusters beforehand.*

4. Experiment

Experimental Setup. We consider three baseline algorithms in our experiments. The Deep-Kernel-based Bayesian Optimization initialized with a pre-trained AutoEncoder (DKBO-AE) applies the deep kernel where a pre-trained AutoEncoder initializes the neural network q. The neural network consists of three hidden layers with 1000, 500, and 50 neurons, and ReLU non-linearity respectively. We use squared exponential kernel as the base kernel, i.e. $k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_{\text{SE}}^2 \exp(-\frac{(\mathbf{x}-\mathbf{x}')^2}{2l})$, for the deep kernel, and Thompson Sampling (Chapelle and Li, 2011) for the acquisition function α. Two other partition-based BO algorithms LA-MCTS (Wang et al., 2020) and TurBO (Eriksson et al., 2019) serve as the baselines. Note that DKBO-AE is used as the subroutine for LA-MCTS, TurBO, and the proposed BALLET. The neural network architecture, base kernel and acquisition function are the same. Therefore the comparison between BALLET and DKBO-AE serves as the ablation study of the proposed partitioning method.

One crucial problem in practice is tuning the hyperparameters. We take the default hyperparameters from the available LA-MCTS [1] and TurBO [2] open-source implementation. Note that we choose TurBO-1 implementation for TurBO where there is one trust region through the optimization, as previous work has shown its robust performance in various tasks (Eriksson et al., 2019).

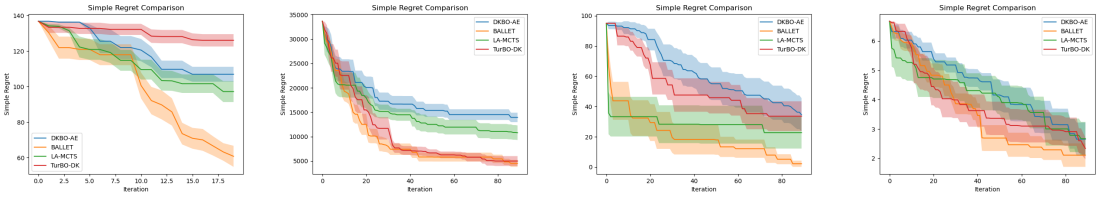
Datasets. *HDBO-200D.* We create a synthetic dataset Sum-200D of 200 dimensions. Each dimension is independently sampled from a standard normal distribution to maximize the uncertainty on that dimension and examine the algorithm's capability to solve the medium-dimensional problem. We want to maximize the label $f(\mathbf{x}) = \sum_{i=1}^{200} e^{x_i}$ which bears an additive structure and of non-linearity. The neural network is pretrained on 100 data points.

Water Converter Configuration-16D. This UCI dataset we use consists of positions and absorbed power outputs of wave energy converters (WECs) from the southern coast of Sydney. The applied converter model is a fully submerged three-tether converter called CETO. 16 WECs locations are placed and optimized in a size-constrained environment.

Nanophotonics Structure Design. We wish to optimize a weighted figure of merit quantifying the fitness of the transmission spectrum for hyperspectral imaging as assessed by a numerical solver (Song et al., 2018). This problem has a 5-dimensional input corresponding to the physical design dimensions of a potential filter. Although the input is not high-dimensional, the function represents a discrete solution of Maxwell's equations and has a complex value landscape.

Rosetta Protein Design. We use a protein engineering dataset describing a set of antigen/antibody binding calculations. These calculations, executed using supercomputing resources, estimate the change in binding free energy at the interface between 71769 modified antibodies and the SARS-CoV-2 spike protein, as compared to a reference antibody. Estimations of binding free energy ($\Delta\Delta G$) are calculated using protein structure based Rosetta Flex simulation software (Das and Baker, 2008; Barlow et al., 2018). These calculations take several CPU hours each and are produced during an antibody design process (Desautels et al., 2020). Inputs are described with an 80-dimensional feature vector that, relative to the reference sequence, describes changes in the interface between the antibody and the corresponding target region on the SARS-CoV-2 spike. This is a particularly relevant problem setting when trying to rapidly choose antibody candidates to respond to a new disease in a timely fashion.

1. https://github.com/facebookresearch/LaMCTS
2. https://botorch.org/tutorials/turbo_1



(a) HDBO-200D (b) Water Converter (c) Nanophotonic (d) Rosetta

Figure 2: Simulation results on each task is shown here. The results on each tasks are collected from at least 10 independent trials. The error bar demonstrates the standard error. β are set to 2 for HDBO-200D, Water Converter and Nanophotonic, and 1 for Rosetta.

Results. As is shown in figure 2, the experiments demonstrate the robust performance of BALLET which consistently matches or outperforms the best baseline. In contrast, LA-MCTS consistently matches or outperforms DKBO-AE, but lags behind both TurBO and DKBO-AE in figure 2b and figure 2d. Note that we also find that using SVM to generalize the partition on Y_t to $\mathbf{X}$ in LA-MCTS occasionally fails possibly due to the intrinsic complexity of the partition learned on Y_t demanding methods of greater capability, while the level-set partition of BALLET is regularized by the smoothness of the global Gaussian process $\mathcal{GP}_{f_h}$. We reject the failed LA-MCTS trials. TurBO matches BALLET performance in figure 2b and figure 2d, but loses to DKBO-AE in figure 2a. By construction, HDBO-200D could have multiple distant local maximum, while TurBO relies on the locality of the observation to identify the trust regions. TurBO is likely to get trapped in the local maximum and the performance degrades in the scenario where the multiple maxima are distant from each other and the gap between sub-optimal and optimal observation is significant. In contrast, BALLET is capable of identifying multiple regions of interest with the level-set partitioning without specifying the desired number of regions.

5. Conclusion

We propose a novel framework for adaptively learning local regions of interest for Bayesian optimization. Our model maintains two Gaussian processes: one global model for identifying the (local) regions of interest as (adaptive) level sets; the other local model for acquiring data in the high-confidence regions of interest. We demonstrate our algorithm in promising real-world experiment design scenarios, including protein engineering and material science. Our results show that BALLET compare favorably against state-of-the-art BO approaches under similar settings—especially in high-dimensional and structured tasks with non-stationary dynamics—while having less hyperparameters to fine-tune.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344 and was supported by the LLNL-LDRD Program under Project No. 20-ERD-032. Lawrence Livermore National Security, LLC. LLNL-CONF-835910-DRAFT. The DOD's Joint Program Executive Office for Chemical, Biological, Radiological and Nuclear Defense (JPEO-CBRND), in collaboration with the Defense Health Agency (DHA) COVID funding initiative for Rapid co-design of manufacturable and efficacious antibody therapeutics for COVID-19 via a machine-learning-driven computational design platform, molecular dynamics simulations, and experimental validation, Lawrence Livermore National Laboratory (LLNL), Proposal L22260, Agreement ID#44208 was used for this effort. Fengxue Zhang was partially supported by NSF #2037026.

References

Kyle A Barlow, Shane O Conchuir, Samuel Thompson, Pooja Suresh, James E Lucas, Markus Heinonen, and Tanja Kortemme. Flex ddg: Rosetta ensemble-based estimation of changes in protein–protein binding affinity upon mutation. *The Journal of Physical Chemistry B*, 122(21): 5389–5399, 2018.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. The curse of dimensionality for local kernel machines. *Techn. Rep*, 1258:12, 2005.

Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with Gaussian processes. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 493–496, 2016. URL https://arxiv.org/abs/1509.01066.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24, 2011.

Rhiju Das and David Baker. Macromolecular modeling with rosetta. *Annu. Rev. Biochem.*, 77: 363–382, 2008.

Thomas Desautels, Adam Zemla, Edmond Lau, Magdalena Franco, and Daniel Faissol. Rapid in silico design of antibodies targeting sars-cov-2 using machine learning and supercomputing. *BioRxiv*, 2020.

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

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 5496–5507, 2019. URL http://papers.nips.cc/paper/8788-scalable-global-optimization-via-local-bayesian-optimization.pdf.

Mafalda Falcão Ferreira, Rui Camacho, and Luís F Teixeira. Using autoencoders as a weight initialization method on deep neural networks for disease detection. *BMC Medical Informatics and Decision Making*, 20(5):1–18, 2020.

Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 1344–1350. AAAI Press, 2013. ISBN 9781577356332. URL https://dl.acm.org/doi/10.5555/2540128.2540322.

Kenji Kawaguchi, Yu Maruyama, and Xiaoyu Zheng. Global continuous optimization with error bound and fast convergence. *Journal of Artificial Intelligence Research*, 56:153–195, 2016. URL https://www.jair.org/index.php/jair/article/view/11007/26166.

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.

Anastasia Makarova, Ilnura Usmanova, Ilija Bogunovic, and Andreas Krause. Risk-averse heteroscedastic bayesian optimization. *Advances in Neural Information Processing Systems*, 34, 2021.

Erich Merrill, Alan Fern, Xiaoli Fern, and Nima Dolatnia. An empirical study of bayesian optimization: Acquisition versus partition. *Journal of Machine Learning Research*, 22(4):1–25, 2021. URL http://jmlr.org/papers/v22/18-220.html.

Rémi Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. *Advances in neural information processing systems*, 24, 2011.

Rémi Munos. *From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning*. now, 2014. URL https://ieeexplore.ieee.org/document/8187198.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

Mikita Sazanovich, Anastasiya Nikolskaya, Yury Belousov, and Aleksei Shpilman. Solving black-box optimization challenge via learning search space partition for local bayesian optimization. In Hugo Jair Escalante and Katja Hofmann, editors, *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 77–85. PMLR, 06–12 Dec 2021. URL https://proceedings.mlr.press/v133/sazanovich21a.html.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016. doi: 10.1109/JPROC.2015.2494218.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *26th Annual Conference on Neural Information Processing Systems 2012*, pages 2951–2959, 2012.

Jialin Song, Yury Tokpanov, Yuxin Chen, Dagny Fleischman, Kate Fountaine, Harry Atwater, and Yisong Yue. Optimizing photonic nanostructures via multi-fidelity gaussian processes. *NeurIPS Workshop on Machine Learning for Molecules and Materials*, 2018.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe bayesian optimization with gaussian processes. In *International conference on machine learning*, pages 4781–4789. PMLR, 2018.

Linnan Wang, Rodrigo Fonseca, and Yuandong Tian. Learning search space partition for black-box optimization using monte carlo tree search. *Advances in Neural Information Processing Systems*, 33:19511–19522, 2020. URL https://proceedings.neurips.cc/paper/2020/file/e2ce14e81dba66dbff9cbc35ecfdb704-Paper.pdf.

Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning*, pages 3627–3635. PMLR, 2017.

Zi Wang, Bolei Zhou, and Stefanie Jegelka. Optimization as estimation with gaussian processes in bandit settings. In *Artificial Intelligence and Statistics*, pages 1022–1031. PMLR, 2016a.

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, 2016b.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain, 09–11 May 2016. PMLR.

Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019.

Fengxue Zhang, Brian Nord, and Yuxin Chen. Learning representation for bayesian optimization with collision-free regularization. *arXiv preprint arXiv:2203.08656*, 2022.