Bayesian Optimization in High-Dimensional Spaces: A Brief Survey

Mohit Malu, Gautam Dasarathy, and Andreas Spanias
SenSIP Center, School of ECEE, Arizona State University, Tempe, AZ
{mmalu,gautamd,spanias}@asu.edu

***Abstract*—Bayesian optimization (BO) has been widely applied to several modern science and engineering applications such as machine learning, neural networks, robotics, aerospace engineering, experimental design. BO has emerged as the *modus operandi* for global optimization of an arbitrary expensive to evaluate black box function f. Although BO has been very successful in low dimensions, scaling it to high dimensional spaces has been significantly challenging due to its exponentially increasing statistical and computational complexity with increasing dimensions. In this era of high dimensional data where the input features are of million dimensions scaling BO to higher dimensions is one of the important goals in the field. There has been a lot of work in recent years to scale BO to higher dimensions, in many of these methods some underlying structure on the objective function is exploited. In this paper, we review recent efforts in this area. In particular, we focus on the methods that exploit different underlying structures on the objective function to scale BO to high dimensions.**

I. Introduction

Several modern science and engineering applications such as machine learning, neural networks, robotics, aerospace engineering, experimental design, require optimization of unknown, expensive to evaluate, black box function within a constrained budget of time and power. Bayesian optimization (BO) is a popular strategy to optimize these expensive to evaluate functions as it provides a sample efficient framework for global optimization as compared to other alternatives such as DIRECT [36], simulated annealing [46], latin hypercubes [62]. More formally, the goal of BO is to optimize an expensive to evaluate, black box objective function $f : \mathcal{X} \to \mathbb{R}$, which can be mathematically formulated as follows:

$$\max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$

where domain $\mathcal{X} \subseteq \mathbb{R}^D$ and typically $D \leq 20$. BO has been widely applied in a variety of optimization applications such as hyperparameter tuning [48,68,78], circuit optimization [57], gait learning in robotics [9,59], aerospace engineering [50], gene design [22], chemical design [24], and animation design [8].

The standard BO algorithm consists of two main components [17]: a statistical model, usually a Gaussian Process (GP) [72], from which the black box objective function is assumed to be sampled, and an acquisition function to efficiently navigate (sample) through the given input space. The key advantages of BO include (i) Sample efficient optimization of expensive to evaluate functions, (ii) Black box optimization which requires minimal knowledge of the function, (iii) Derivative free optimization unlike gradient descent. A comprehensive review of BO was done by Shahriari et al. [76], and Brochu et al. [7]. The extensions of standard BO with derivative information [14,95], multi-fidelity optimization [32,38,39,41,44,81], trust region based BO [15], and neural network architecture search [40,48] have also been studied.

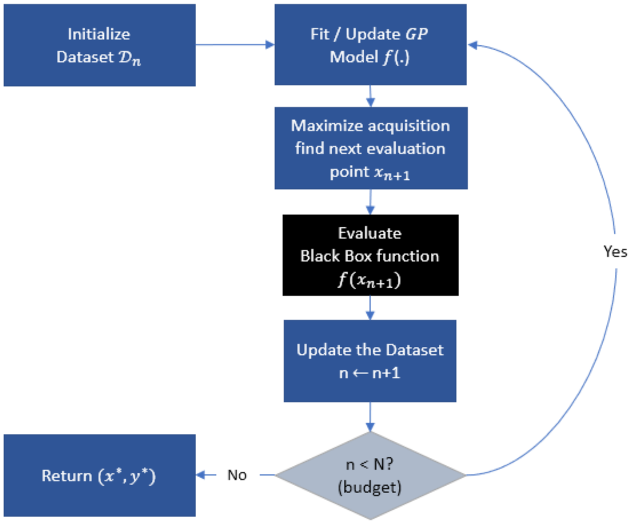
Initialize Dataset $\mathcal{D}_n$
Fit / Update *GP* Model $f(.)$
Maximize acquisition find next evaluation point x_{n+1}
Evaluate Black Box function $f(x_{n+1})$
Update the Dataset n ← n+1
n < N? (budget)
Return (x^*, y^*)
No
Yes

Fig. 1. A flow graph for Bayesian Optimization.

Although BO has been very successful in moderate to low dimensions, scaling BO to high dimensional space is significantly challenging due to its exponentially increasing statistical and computational complexity with increasing dimensions. More specifically, (i) As the dimensionality of the objective function increases, number of points (queries) required to cover the input space increases exponentially. (ii) BO mandates finding the maximizer of the acquisition function, which in itself is a non-convex optimization problem over the input space that requires exponentially increasing computational power with increasing dimensionality. These factors inhibit the direct adoption of BO to high dimensional optimization.

Algorithm 1 Bayesian Optimization

```
Place a Bayesian prior on $f$ (Typically Gaussian)
Observe $f$ at $n_0 > 0$ points.
Set $n = n_0, \mathcal{D}_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
while $n \leq N$ do
    Compute the posterior $f|\mathcal{D}_n$ using Bayes Theorem
    Compute the acquisition function $a_n(\mathbf{x})$ based on the posterior.
    Find $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x} \in \mathcal{X}} a_n(\mathbf{x})$.
    Observe $y_{n+1} = f(\mathbf{x}_{n+1}) + \epsilon_{n+1}$. ($\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, \eta^2)$)
    $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$
    increment $n$
end while
Return the point $\mathbf{x}_i$ with largest $y_i$
```

Scaling BO to high dimensions is gaining more attentions in the field because in this age of high dimensional data and increasingly complex systems, high dimensional optimization problems are ubiquitous. For example, large scale hyper-parameter tuning of neural networks [90], biology [22], computational astrophysics [69], and computer vision [5]. There has been a lot of effort in recent years to extend BO to higher dimensions, most of these works assume some underlying structure on the objective function. The structural assumptions constitute of two types: (i) Intrinsic low dimensionality of the objective function (ii) Additive structure i.e., decomposition of the objective function into sum of low dimensional functions. In this paper, we perform a comprehensive review of the existing approaches for high dimensional BO. In particular, we focus on the methods that exploit different underlying structures on the objective function to scale BO to high dimensions.

The rest of the paper is organized as follows, in Section II we present an overview of Bayesian Optimization (BO), followed by challenges and methods to scale BO to high dimensions in section III and IV. Section V discusses the applications of BO, and the conclusion in section VI.

II. Overview of Bayesian Optimization

Bayesian Optimization is a sequential optimization technique with two key components: (i) a Bayesian statistical model for modeling the objective function f, and (ii) an acquisition function to decide where to sample next in the domain $\mathcal{X}$. Algorithm 1 above gives an overview of a typical BO procedure.

A. Bayesian Statistical Model - Gaussian Process

The statistical model translates the information gained from previous observations onto the entire domain of the function by forming a posterior over the objective function conditioned over the data. The posterior is updated every time the function is evaluated (queried). Typically, Gaussian processes (GP) are the go to statistical model for modeling objective function f, as GP offers flexibility in terms of it providing closed form solutions while evaluating the posterior distribution. Formally,

$$f \sim \mathcal{GP}(\mu(.), \kappa(., .)).$$

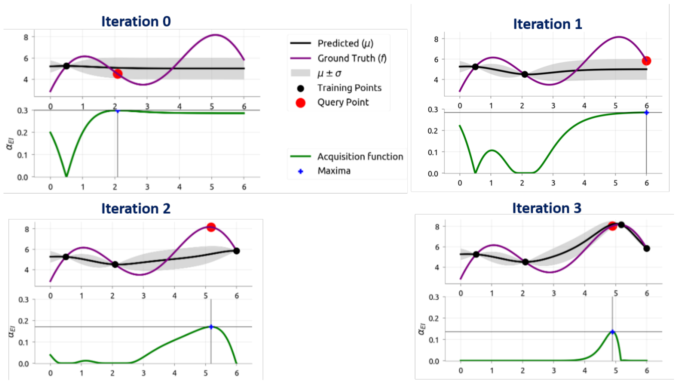

Fig. 2. A pictorial overview of Bayesian Optimization. The posterior mean (black curve) and the standard deviation from the GP regression is used to formulate the acquisition function (green curve). As the iterations increase, it can be observed that BO attempts to reach the maxima of the objective function f. Image source: https://distill.pub/2020/bayesian-optimization/.

where the GP is completely given by its mean $\mu(.)$ and covariance $\kappa(., .)$ functions. Let $\mathcal{D}_n : \{(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_n, y_n)\}$ be the initial set of data samples (noisy observations) of the true objective function f. The posterior of the function f on the observed data $\mathcal{D}_n$ is also Gaussian. i.e. $f(\mathbf{x})|\mathcal{D}_n \sim \mathcal{N}(\mu_n(\mathbf{x}), \sigma_n^2(\mathbf{x}))$, where the mean $\mu_n(\mathbf{x})$ and covariance $\sigma_n^2(\mathbf{x})$ are given as follows:

$$\mu_n(\mathbf{x}) = \mu(\mathbf{x}) + k^T(K + \eta^2 I_n)^{-1}Y$$
$$\sigma_n^2(\mathbf{x}) = \kappa(\mathbf{x}, \mathbf{x}) - k^T(K + \eta^2 I_n)^{-1}k$$

Here, Y is the vector of observations, k is a vector with $k_i = \kappa(\mathbf{x}, \mathbf{x}_i)$. The matrix K is such that $K_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ $i, j \in \{1, \ldots, n\}$. Choice of covariance kernel function mainly depends on the degree of smoothness warranted for the modeled function and is most often selected to be square exponential (SE) kernel or Matérn kernel.

B. Acquisition Function

Acquisition function utilizes the information from previously observed data to navigate through the domain. Typical choice of acquisition function includes Upper Confidence Bound (UCB) [2,79] and Expected Improvement (EI) [63].The acquisition functions quantify the potential of finding a maximizer of the objective function in the entire domain. Hence, in every iteration of BO the function f is queried at that maximizer of the acquisition function. Acquisition function is usually formulated based on the parameters of the posterior distribution. The *Upper confidence Bound* (UCB) is defined as follows:

$$a_n(\mathbf{x}) := \mu_n(\mathbf{x}) + \beta^{1/2}\sigma_n(\mathbf{x})$$

Here, β is a hyperparameter that controls the trade-off between exploration - Domain of f where f has high variance $\sigma_n(\mathbf{x})^2$ and exploitation - Domain of f where f has a high mean $\mu_n(\mathbf{x})$. Figure 2 demonstrates the BO algorithm. As it can be observed, the acquisition function provides the samples from the domain of f that either provide the maximum information (high uncertainty) about the function or is close to optimal (high mean). The maximization of acquisition function is assumed to be an inexpensive task and is optimized using off the shelf methods such as Dividing Rectangles (DIRECT) algorithm [36], CMA-ES [27], L-BFGS [56]. In the following section we will look at the challenges in scaling BO to high dimensions.

III. Challenges in Scaling BO to High Dimensions

BO has been a successful approach for finding the global optima of an unknown function f when the dimensionality of domain of is moderate or low. However, scaling it to higher dimensions is challenging due to its increasing statistical and computational complexity with increase in dimensions. The reasons for this are discussed in detail below:

1) Coverage of the Domain: To ensure that the global optimum is found, we require good coverage of the domain of f, but as dimensionality D increases, the number of evaluations needed to effectively cover the domain increase exponentially.

2) Computation and Storage Cost: The computation of posterior requires the computation of the inverse of the kernel matrix, whose size depends on the number of function evaluations made so far i.e., $\mathcal{O}(N^3)$, where N is the number of evaluations. Hence, as the number of evaluations of the function increase exponentially with increase in dimensions, computational and storage costs for computing posterior distribution also increases exponentially with dimensions.

3) Maximizing the Acquisition Function: Acquisition function is maximized by using standard global optimization algorithms such as DIRECT or L-BFGS. These methods have query complexity that exponentially depends on the dimensions D i.e., it requires $\mathcal{O}(\zeta^{-D})$ iterations to maximize the function within ζ accuracy. Hence, these algorithms work efficiently only when the input space is moderate or low dimensional. However, increase in dimensionality of the input leads to slower convergence.

4) Function Estimation: Non-parametric regression becomes difficult in high dimensions. Provable lower bounds in [26] demonstrate the exponential dependence on the dimension D. This is often referred to as *curse of dimensionality* for non-parametric regression problems.

IV. Methods to Scale BO to High Dimensions

There has been a lot of effort in recent years to scale BO to high dimensions. In this section we'll look at these methods that overcome the said challenges either by exploiting the structure of the objective function or by trading off between convergence and computational complexity.

A. Exploiting Structure

BO can be scaled to higher dimensions by imposing reasonable structural assumptions on the objective functions such as low dimensionality or additive structure.

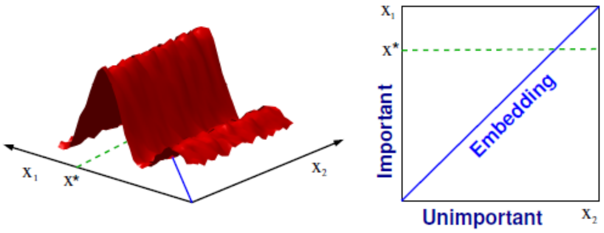

Fig. 3. The function in D=2 dimensions only has one effective (active) dimension. Hence performing BO on the random embedding can still find the optimum [89].

1) Intrinsic Low Dimensionality: In many applications, the objective function depends only on low dimensional subspace, examples include hyperparameter optimization for neural networks [4] and automatic configuration of a mixed integer linear programming solver [33].

Wang et al. [89] laid the foundations for BO with random linear embeddings and came up with an algorithm 2 random embedding Bayesian Optimization (REMBO) that exploits the low effective dimensionality of the objective function. The intuition behind the algorithm lies mainly in proving that there exists a low dimensional vector $\mathbf{y}^* \in \mathbb{R}^d$ with probability 1 such that $f(\mathbf{x}^*) = f(\mathbf{A}\mathbf{y}^*)$ for a random projection matrix $\mathbf{A} \in \mathbb{R}^{D\times d}$. This allows BO to be performed in the low dimensional subspace to identify the next point of evaluation which is then projected back to original dimensions using a random projection matrix to evaluate the objective function.

Although REMBO performs well, it can sometimes perform poorly even for some synthetic problems due to over-exploration of boundary and distortions in embedding due to projections from low dimensions to high dimensions lying outside the box bounds. Letham et al. [52] addressed these drawbacks and presents a new algorithm called adaptive linear embedding Bayesian Optimization (ALEBO) which defines an embedding matrix that re-projects the points from embedded space to original space and uses the pseudo inverse to project the bounds in the original space to the embedded space, thereby, avoiding the distortions from clipping of projected points.

Some other works that exploit the low dimensionality to scale BO include: (i) Subspace Identification Bayesian Optimization (SI-BO) [12] which involves estimation of the subspace on which the function is supported using low rank matrix recovery [87] and executes BO on the learned subspace. (ii) Sliced Inverse Regression Bayesian Optimization (SIR-BO) [102] which utilizes sliced inverse regression technique for dimension reduction [53] to find the effective subspace and performs BO on the learned effective subspace, this method is also extended to nonlinear dimension reduction using kernel trick. (iii) Hashing enhanced Subspace BO (HeSBO) [67] performs BO on the low dimensional space and uses two hash functions [10] to construct the inverse subspace embedding to recover the vector in original space from the low dimensional

Algorithm 2 REMBO

```
Generate a random matrix $\mathbf{A}$
Place a Gaussian prior on $g(\mathbf{y}) = f(\mathbf{A}\mathbf{y})$
Choose the domain $\mathcal{Y}$ such that $\mathbf{y} \in \mathcal{Y}$
for $n = 1, 2, \ldots$ do
    Compute the posterior $f|\mathcal{D}_n$
    Find $\mathbf{y}_{n+1} \in \mathbb{R}^d$ such that $\mathbf{y}_{n+1} = \arg\max_{\mathbf{y} \in \mathcal{Y}} a_n(\mathbf{y})$.
    $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\mathbf{y}_{t+1}, f(\mathbf{A}\mathbf{y}_{t+1}))\}$
    increment $n$
end for
```

vectors. (iv) Non linear embedding method [65] that learns the nonlinear embedding using the feed forward neural network to reduce the dimensionality of the input, applies BO with manifold Gaussian Process in the embedding space and then uses the reconstruction mapping generated using multi-output GP to project back to the original space for function evaluation.

2) Additive Structure: A promising alternative to scale BO to high dimension that can also model richer class of functions as compared to low dimensionality assumption was proposed by Kandasamy et al. [37] which adopts to the method introduced by Duvenaud et al. [13]. Here, the objective function $f : \mathcal{X} \to \mathbb{R}$ w.l.o.g $\mathcal{X} = [0,1]^D$ with additive structure decomposes as the sum of independent low dimensional functions each of which depends on disjoint subset of dimensions i.e.,

$$f(\mathbf{x}) = f^{(1)}(\mathbf{x}^{(1)}) + f^{(2)}(\mathbf{x}^{(2)}) + \cdots + f^{(M)}(\mathbf{x}^{(M)}) \qquad (1)$$

where, each $\mathbf{x}^{(j)} \in \mathcal{X}^{(j)} = [0,1]^{d_j}$ are disjoint low dimensional components (groups) and $d^j \leq d \ll D$. Each of these low dimensional function $f^{(j)}$ is assumed to be sampled from a Gaussian Process, $\mathcal{GP}(\mu^{(j)}, \kappa^{(j)})$ where the $f^{(j)}$'s are independent. This implies that f itself is sampled from a Gaussian Process, $\mathcal{GP}(\mu, \kappa)$ where,

$$\mu(\mathbf{x}) = \mu^{(1)}(\mathbf{x}^{(1)}) + \cdots + \mu^{(M)}(\mathbf{x}^{(M)})$$
$$\kappa(\mathbf{x}, \mathbf{x}') = \kappa^{(1)}(\mathbf{x}^{(1)}, \mathbf{x}^{(1)'}) + \cdots + \kappa^{(M)}(\mathbf{x}^{(M)}, \mathbf{x}^{(M)'}).$$

Given the above structure, the paper [37] proposes an alternative acquisition function *Additive Gaussian Process Upper Confidence Bound* (**ADD-GP-UCB**) which applies to additive kernel and is defined as follows:

$$\widetilde{a}_n(\mathbf{x}) = \mu_n(\mathbf{x}) + \beta^{1/2} \sum_{j=1}^{M} \sigma_n^{(j)}(\mathbf{x}^{(j)}). \qquad (2)$$

This can further be written as a sum of functions on orthogonal domains: $\widetilde{a}_n(\mathbf{x}) = \sum_j \widetilde{a}_n^{(j)}(\mathbf{x}^{(j)})$ where $\widetilde{a}_n^{(j)}(\mathbf{x}^{(j)}) = \mu_n^{(j)}(\mathbf{x}^{(j)}) + \beta^{1/2}\sigma_n^{(j)}(\mathbf{x}^{(j)})$. Hence, $\widetilde{a}_n$ can be maximized by maximising each $\widetilde{a}_n^{(j)}$ separately on $\mathcal{X}^{(j)}$.

The ADD-GP-UCB algorithm (Algorithm 3) has two main components. First, compute the posterior for each $f^{(j)}$, then maximize an atmost d dimension GP-UCB like acquisition function on each low dimensional GP to construct the next query point. Though this approach models the richer class of functions, it is still restrictive in terms of requiring the decomposition to be axis aligned. Li et al. [54] addressed this by considering the additive model on the projected data [21]. Hence, generalizing the additive assumption to projected additive assumption [28].

Algorithm 3 ADD-GP-UCB

```
Kernels $\kappa^{(1)}, \ldots, \kappa^{(M)}$, Decomposition $(\mathcal{X}^{(j)})_{j=1}^{M}$
Place a Gaussian prior on each $f^{(j)} \sim \mathcal{GP}(\mathbf{0}, \kappa^{(j)})$
for $n = 1, 2, \ldots$ do
    for j = 1,...,M do
        Compute the posterior $f^{(j)}|\mathcal{D}_n$
        $\mathbf{x}_{n+1}^{(j)} = \arg\max_{z \in \mathcal{X}^{(j)}} \widetilde{a}_n^{(j)}(\mathbf{x}^{(j)})$
    end for
    $\mathbf{x}_{n+1} = \cup_{j=1}^{M} \mathbf{x}_{n+1}^{(j)}$
    $\mathbf{y}_{n+1} = f(\mathbf{x}_{n+1}) + \epsilon_{n+1}$
    $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\mathbf{x}_{n+1}, \mathbf{y}_{n+1}\}$
    increment $n$
end for
```

The performance of the mentioned additive model based algorithms depend on the knowledge of the groups of decomposition and learning them is computationally challenging. [37,54] learn the decomposition by randomly sampling the decompositions and selecting the one that maximizes the marginal likelihood, whereas [19,91] introduce efficient approaches to learn decompositions based on Gibbs sampling and Markov Chain Monte Carlo (MCMC) methods respectively.

The additive model with disjoint decomposition was further generalized to have the overlapping groups by Rolland et al., [73] and Hoang et al., [31] where the overlapping decomposition is represented by a dependency graph or sparse factor graph and acquisition function is optimized by using message passing protocol. Further work to improve on additive models include batched BO techniques by [91,92], and deterministic fourier feature approximation of the stationary kernel for efficient optimization by [66].

B. Non-Structure based Methods

Li et al. [55] explored dropout strategy (similar to the dropout strategy used in neural network hyperparameter tuning) to scale BO to high dimensions, wherein BO is performed over randomly selected d out of D dimensions in every iteration, and rest of the dimensions are filled using different strategies such as random values, values from the best found solution so far and a mixture of random and best values. The theoretical analysis and experimental results show that the regret gap increases with the increase in the number of parameters dropped. Hence, this algorithm trades off between computational complexity and convergence to the optimal solution.

Gupta et al. [25] proposed a method on the similar idea as that of dropout strategy [55], wherein the acquisition function is maximized over the restricted space consisting of multiple low dimensional subspaces of the high dimensional space. Theoretical analysis of the algorithm shows that the bounds on the cumulative regret gets tighter as the number of subspaces increase, in turn increasing the computational complexity.

Rana et al. [70] proposed an elastic Gaussian Process model to efficiently traverse through zero gradient regions while optimizing acquisition function in high dimensions. Though this gives an improved solution for acquisition optimization, it still does not tackle other scalability issues of BO.

V. Applications in Machine Learning and Other Areas

Bayesian Optimization has been employed in several applications including machine learning, deep learning, autonomous vehicles, biomedical and radar. In the following we briefly describe select applications of BO with the emphasis on machine learning.

1) Machine Learning: While machine learning algorithms have been covered in survey papers [77,80] and text books [3,6,83,85], our coverage here focuses specially on methods where BO is part of the model hyperparameter estimation process.

More specifically we concentrate on hyperparameter tuning in machine learning and deep learning applications. For example, [78] discussed the automatic tuning of hyperparameter in the BO framework, where the generalization performance of machine learning or deep learning algorithm can be viewed as a black box function with hyperparameters as the input. The application of BO to this function leads to finding an optimal set of hyperparameters to better generalize the model. Figure 4 shows flow graph of application of BO for hyperparameter tuning. [48] proposed a fast hyperparameter tuning method that uses multi task BO technique [81] on SVM and CNN models to select the hyperparameters and also subset of entire data that yields the most information about the performance of the given algorithm configuration on entire data. This reduces the model training time which in turn reduces overall hyperparameter tuning time. [11] provided a guide for using BO techniques in machine learning. They also apply BO for tuning hyperparameters of CNN for image classification and show that BO outperforms the random search over the parameter space. [96] applied BO to random forests and neural networks, and empirically show the superior performance of BO as compared to random search and brute force methods. [94] proposed a promising strategy for network architecture search using BO coupled with neural predictor. Some other machine learning applications of BO include accelerated hyperparameter search [68,82], automated hyperparameter tuning [88], tuning of network security based machine learning models [35], and hyperparameter optimization in computer vision architecture [5].

2) Circuit Design: Circuit design / optimization involves simulation of multiple circuits which are computationally intensive for large scale complicated circuits. [57] proposed a weighted expected improvement based BO for automated circuit optimization. [86] utilized BO techniques to meet or exceed design specifications of high performance systems. [103] proposed a neural net based BO for analog circuit synthesis.

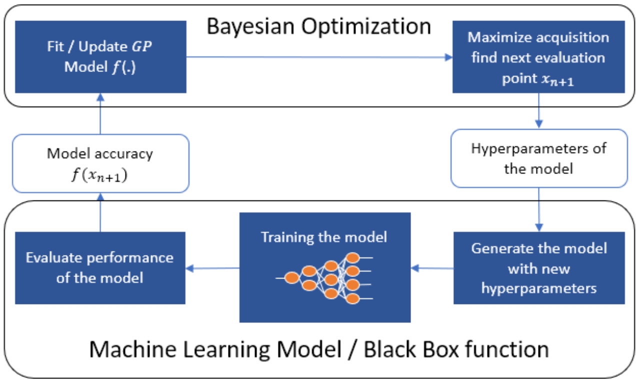
Bayesian Optimization
Fit / Update *GP* Model $f(.)$
Maximize acquisition find next evaluation point x_{n+1}
Hyperparameters of the model
Model accuracy $f(x_{n+1})$
Evaluate performance of the model
Training the model
Generate the model with new hyperparameters
Machine Learning Model / Black Box function

Fig. 4. A flow graph of hyperparameter tuning of machine learning model using BO.

3) Robotics: Many optimization problems in robotics can be tackled with BO framework. For example, [9] utilized BO for gait design and control, [59] proposed BO approach for online path planning for optimal sensing, and [60] applied BO with kernel functions for active policy search for robot control.

4) Health Related Applications: Chemical / drug synthesis involves various optimization tasks which typically require numerous experiments to obtain the optimal formulation, and we can reduce the number of experiments required using BO framework. [75] introduced application of BO for drug synthesis, [49] proposed ChemBO: a BO framework for generating and optimizing organic molecule with desired properties, [24] used constrained BO along with variational auto-encoders for automatic chemical design, [30] used BO to tune the parameters of the deep convolutional neural network (DCNN) for computer aided diagnosis scheme for distinguishing between benign and malignant masses.

5) Other Applications: Some other applications of BO include synthetic gene design [22], design of aerospace engineering systems [50], animation design [8], computational astrophysics [69], Transformer language models for speech recognition [98], material design [16], and experimental Design [23,34].

More reading and bibliography on applications of BO is given in the following: solar energy [71,93], autonomous vehicles [18], radar systems [51,97], data science [1,47], Internet of Things [42,58], sensors [20,101], health and wearable [43,45], stochastic optimization with adaptive restarts [61], multi-fidelity modelling in global optimization approaches [99], surveillance [29,84], environmental systems [64,74].

VI. Conclusion

In this review, we present an overview of Bayesian Optimization(BO) a sample efficient framework for optimizing expensive to evaluate black box objective function and it's increasing applications in recent years. We also briefly discuss the statistical and computational challenges faced in scaling BO to high dimensions and focused on recent efforts to over come the said challenges. Further, we discuss various algorithms that exploit structure, dwelling more into the details of REMBO that exploits intrinsic low dimensionality and, ADD-GP-UCB which exploits additive structure for the objective function. We also give brief description of the algorithms that do not assume any structure on the objective function instead trade convergence for computational complexity to scale BO to higher dimensions. Finally, we conclude by describing select real world applications of BO with emphasis on machine learning applications.

Some possible directions of future work in this field are: (i) Exploiting graph structure to scale BO to high dimensions which would be useful under scenarios where the objective function has an underlying graph structure. Examples of such functions include traffic patterns as a function of road network. (ii) High dimensional multi-output BO, where few of the ideas from high dimensional BO can be applied to scale multi-output Bayesian Optimization to higher dimensions.

Acknowledgment

This project was supported in part by the NSF Awards 1540040, 2003111, 2048223 and the SenSIP Center.

References

[1] F. Archetti and A. Candelieri, Bayesian optimization and data science. Springer, 2019.

[2] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," Journal of Machine Learning Research, vol. 3, no. Nov, pp. 397–422, 2002.

[3] D. Barber, Bayesian reasoning and machine learning. Cambridge University Press, 2012.

[4] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," Journal of machine learning research, vol. 13, no. 2, 2012.

[5] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in International conference on machine learning, 2013, pp. 115–123.

[6] C. M. Bishop, Pattern recognition and machine learning. springer, 2006.

[7] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," arXiv preprint arXiv:1012.2599, 2010.

[8] E. Brochu, T. Brochu, and N. de Freitas, "A Bayesian interactive optimization approach to procedural animation design," in Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2010, pp. 103–112.

[9] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," Annals of Mathematics and Artificial Intelligence, vol. 76, no. 1, pp. 5–23, 2016.

[10] M. Charikar, K. Chen, and M. Farach-Colton, "Finding frequent items in data streams," Theoretical Computer Science, vol. 312, no. 1, pp. 3–15, 2004.

[11] I. Dewancker, M. McCourt, and S. Clark, "Bayesian optimization for machine learning: A practical guidebook," arXiv preprint arXiv:1612.04858, 2016.

[12] J. Djolonga, A. Krause, and V. Cevher, "High-dimensional Gaussian process bandits," in Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1, 2013, pp. 1025–1033.

[13] D. K. Duvenaud, H. Nickisch, and C. Rasmussen, "Additive Gaussian Processes," Advances in Neural Information Processing Systems, vol. 24, pp. 226–234, 2011.

[14] D. Eriksson, K. Dong, E. H. Lee, D. Bindel, and A. G. Wilson, "Scaling Gaussian process regression with derivatives," arXiv preprint arXiv:1810.12283, 2018.

[15] D. Eriksson, M. Pearce, J. R. Gardner, R. Turner, and M. Poloczek, "Scalable global optimization via local bayesian optimization," arXiv preprint arXiv:1910.01739, 2019.

[16] P. I. Frazier and J. Wang, "Bayesian optimization for materials design," in Information Science for Materials Discovery and Design, Springer, 2016, pp. 45–75.

[17] P. I. Frazier, "A tutorial on Bayesian optimization," arXiv preprint arXiv:1807.02811, 2018.

[18] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings, "Identification of test cases for automated driving systems using Bayesian optimization," in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 1961–1967.

[19] J. Gardner, C. Guo, K. Weinberger, R. Garnett, and R. Grosse, "Discovering and Exploiting Additive Structure for Bayesian Optimization," in Artificial Intelligence and Statistics, Apr. 2017, pp. 1311–1319.

[20] R. Garnett, M. A. Osborne, and S. J. Roberts, "Bayesian optimization for sensor set selection," in Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks, 2010, pp. 209–219.

[21] E. Gilboa, Y. Saatçi and J. Cunningham, "Scaling Multidimensional Gaussian Processes using Projected Additive Approximations," in International Conference on Machine Learning, Feb. 2013, pp. 454–461.

[22] J. Gonzalez, J. Longworth, D. C. James, and N. D. Lawrence, "Bayesian optimization for synthetic gene design," arXiv preprint arXiv:1505.01627, 2015.

[23] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh, "Bayesian optimization for adaptive experimental design: A review," IEEE Access, vol. 8, pp. 13937–13948, 2020.

[24] R.-R. Griffiths and J. M. Hernández-Lobato, "Constrained bayesian optimization for automatic chemical design," arXiv preprint arXiv:1709.05501, 2017.

[25] S. Gupta, S. Rana, and S. Venkatesh, "Trading Convergence Rate with Computational Budget in High Dimensional Bayesian Optimization," in Proceedings of the AAAI Conference on Artificial Intelligence, 2020, vol. 34, no. 03, pp. 2425–2432.

[26] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk, A distribution-free theory of nonparametric regression, vol. 1. Springer, 2002.

[27] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," Evolutionary computation, vol. 9, no. 2, pp. 159–195, 2001.

[28] T. J. Hastie and R. J. Tibshirani, "Generalized additive models london chapman and hall," Inc, 1990.

[29] J. Henrio, T. Deligne, T. Nakashima, and T. Watanabe, "Route planning for multiple surveillance autonomous drones using a discrete firefly algorithm and a Bayesian optimization method," Artificial Life and Robotics, vol. 24, no. 1, pp. 100–105, 2019.

[30] A. Hizukuri, R. Nakayama, M. Nara, M. Suzuki, and K. Namba, "Computer-Aided Diagnosis Scheme for Distinguishing Between Benign and Malignant Masses on Breast DCE-MRI Images Using Deep Convolutional Neural Network with Bayesian Optimization," Journal of Digital Imaging, vol. 34, no. 1, pp. 116–123, 2021.

[31] T. N. Hoang, Q. M. Hoang, R. Ouyang, and K. H. Low, "Decentralized high-dimensional Bayesian optimization with factor graphs," in Proceedings of the AAAI Conference on Artificial Intelligence, 2018, vol. 32, no. 1.

[32] D. Huang, T. T. Allen, W. I. Notz, and R. A. Miller, "Sequential kriging optimization using multiple-fidelity evaluations," Structural and Multidisciplinary Optimization, vol. 32, no. 5, pp. 369–382, 2006.

[33] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Automated configuration of mixed integer programming solvers," in International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming, 2010, pp. 186–202.

[34] M. Imani and S. F. Ghoreishi, "Bayesian optimization objective-based experimental design," in 2020 American Control Conference (ACC), 2020, pp. 3405–3411.

[35] M. Injadat, F. Salo, A. B. Nassif, A. Essex, and A. Shami, "Bayesian optimization with machine learning algorithms towards anomaly detection," in 2018 IEEE global communications conference (GLOBECOM), 2018, pp. 1–6.

[36] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," Journal of optimization Theory and Applications, vol. 79, no. 1, pp. 157–181, 1993.

[37] K. Kandasamy, J. Schneider, and B. Póczos, "High dimensional Bayesian optimisation and bandits via additive models", in International conference on machine learning, 2015, pp. 295–304.

[38] K. Kandasamy, G. Dasarathy, J. Oliva, J. Schneider, and B. Póczos, "Gaussian process bandit optimisation with multi-fidelity evaluations," in

Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 1000–1008.

[39] K. Kandasamy, G. Dasarathy, J. Schneider, and B. Póczos, “Multi-fidelity bayesian optimisation with continuous approximations,” in International Conference on Machine Learning, 2017, pp. 1799–1808.

[40] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. Xing, “Neural architecture search with bayesian optimisation and optimal transport,” arXiv preprint arXiv:1802.07191, 2018.

[41] K. Kandasamy, G. Dasarathy, J. Oliva, J. Schneider, and B. Poczos, “Multi-fidelity gaussian process bandit optimisation,” Journal of Artificial Intelligence Research, vol. 66, pp. 151–196, 2019.

[42] L. Kang, R.-S. Chen, N. Xiong, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, “Selecting hyper-parameters of Gaussian process regression based on non-inertial particle swarm optimization in Internet of Things,” IEEE Access, vol. 7, pp. 59504–59513, 2019.

[43] H. Ke et al., “Improving brain E-health services via high-performance EEG classification with grouping Bayesian optimization,” IEEE Transactions on Services Computing, vol. 13, no. 4, pp. 696–708, 2019.

[44] M. C. Kennedy and A. O’Hagan, “Predicting the output from a complex computer code when fast approximations are available,” Biometrika, vol. 87, no. 1, pp. 1–13, 2000.

[45] M. Kim et al., “Human-in-the-loop Bayesian optimization of wearable device parameters,” PloS one, vol. 12, no. 9, p. e0184054, 2017.

[46] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” science, vol. 220, no. 4598, pp. 671–680, 1983

[47] A. Klein, S. Bartels, S. Falkner, P. Hennig, and F. Hutter, “Towards efficient Bayesian optimization for big data,” 2015.

[48] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, “Fast bayesian optimization of machine learning hyperparameters on large datasets,” in Artificial Intelligence and Statistics, 2017, pp. 528–536.

[49] K. Korovina et al., “Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations,” in International Conference on Artificial Intelligence and Statistics, 2020, pp. 3393–3403.

[50] R. Lam, M. Poloczek, P. Frazier, and K. E. Willcox, “Advances in bayesian optimization with applications in aerospace engineering,” in 2018 AIAA Non-Deterministic Approaches Conference, 2018, p. 1656.

[51] H. T. Le, S. L. Phung, A. Bouzerdoum, and F. H. C. Tivive, “Human motion classification with micro-Doppler radar and bayesian-optimized convolutional neural networks,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 2961–2965.

[52] B. Letham, R. Calandra, A. Rai, and E. Bakshy, “Re-examining linear embeddings for high-dimensional Bayesian optimization,” Advances in Neural Information Processing Systems, vol. 33, 2020.

[53] K.-C. Li, “Sliced inverse regression for dimension reduction,” Journal of the American Statistical Association, vol. 86, no. 414, pp. 316–327, 1991.

[54] C.-L. Li, K. Kandasamy, B. Póczos, and J. Schneider, “High dimensional Bayesian optimization via restricted projection pursuit models,” in Artificial Intelligence and Statistics, 2016, pp. 884–892.

[55] C. Li, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, and A. Shilton, “High dimensional Bayesian optimization using dropout,” in Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 2096–2102.

[56] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” Mathematical programming, vol. 45, no. 1, pp. 503–528, 1989.

[57] W. Lyu et al., “An efficient bayesian optimization approach for automated optimization of analog circuits,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 65, no. 6, pp. 1954–1967, 2017.

[58] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, “Machine learning for Internet of Things data analysis: A survey,” Digital Communications and Networks, vol. 4, no. 3, pp. 161–175, 2018.

[59] R. Martinez-Cantin, N. De Freitas, E. Brochu, J. Castellanos, and A. Doucet, “A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot,” Autonomous Robots, vol. 27, no. 2, pp. 93–103, 2009.

[60] R. Martinez-Cantin, “Bayesian optimization with adaptive kernels for robot control,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3350–3356.

[61] L. Mathesen, G. Pedrielli, S. H. Ng, and Z. B. Zabinsky, “Stochastic optimization with adaptive restart: A framework for integrated local and global learning,” Journal of Global Optimization, vol. 79, no. 1, pp. 87–110, 2021.

[62] M. D. McKay, R. J. Beckman, and W. J. Conover, “Comparison the three methods for selecting values of input variable in the analysis of output from a computer code,” Technometrics;(United States), vol. 21, no. 2, 1979.

[63] J. Mockus, “Application of Bayesian approach to numerical methods of global and stochastic optimization,” Journal of Global Optimization, vol. 4, no. 4, pp. 347–365, 1994.

[64] P. Morere, R. Marchant, and F. Ramos, “Sequential Bayesian optimization as a POMDP for environment monitoring with UAVs,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 6381–6388.

[65] R. Moriconi, M. P. Deisenroth, and K. S. Kumar, “High-dimensional Bayesian optimization using low-dimensional feature spaces,” Machine Learning, vol. 109, no. 9, pp. 1925–1943, 2020.

[66] M. Mutný and A. Krause, “Efficient high dimensional bayesian optimization with additivity and quadrature fourier features,” Advances in Neural Information Processing Systems 31, pp. 9005–9016, 2019.

[67] A. Nayebi, A. Munteanu, and M. Poloczek, “A framework for Bayesian optimization in embedded subspaces,” in International Conference on Machine Learning, 2019, pp. 4752–4761.

[68] V. Nguyen, “Bayesian optimization for accelerating hyper-parameter tuning,” in 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), 2019, pp. 302–305.

[69] D. Parkinson, P. Mukherjee, and A. R. Liddle, “Bayesian model selection analysis of WMAP3,” Physical review D, vol. 73, no. 12, p. 123523, 2006.

[70] S. Rana, C. Li, S. Gupta, V. Nguyen, and S. Venkatesh, “High dimensional Bayesian optimization with elastic Gaussian process”, *International conference on machine learning*, 2017, pp. 2883–2891.

[71] S. Rao et al., “Machine Learning for Solar Array Monitoring, Optimization, and Control,” Synthesis Lectures on Power Electronics, vol. 7, no. 1, pp. 1–91, 2020.

[72] C. E. Rasmussen, “Gaussian processes in machine learning,” in Summer school on machine learning, 2003, pp. 63–71.

[73] P. Rolland, J. Scarlett, I. Bogunovic, and V. Cevher, “High-dimensional Bayesian optimization via additive models with overlapping groups,” in International conference on artificial intelligence and statistics, 2018, pp. 298–307.

[74] F. P. Samaniego, D. G. Reina, S. L. T. Marín, M. Arzamendia, and D. O. Gregor, “A Bayesian optimization approach for water resources monitoring through an autonomous surface vehicle: The Ypacarai lake case study,” IEEE Access, vol. 9, pp. 9163–9179, 2021.

[75] S. Sano, T. Kadowaki, K. Tsuda, and S. Kimura, “Application of Bayesian optimization for pharmaceutical product development,” Journal of Pharmaceutical Innovation, vol. 15, no. 3, pp. 333–343, 2020.

[76] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, “Taking the human out of the loop: A review of Bayesian optimization,” Proceedings of the IEEE, vol. 104, no. 1, pp. 148–175, 2015.

[77] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, “A brief survey of machine learning methods and their sensor and IoT applications,” in 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA), 2017, pp. 1–8.

[78] J. Snoek, H. Larochelle, and R. P Adams. Practical Bayesian Optimization of Machine Learning Algorithms.In NIPS, 2012

[79] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, “Gaussian process optimization in the bandit setting: no regret and experimental design,” in Proceedings of the 27th International Conference on International Conference on Machine Learning, 2010, pp. 1015–1022.

[80] S. Sun, “A survey of multi-view machine learning,” Neural computing and applications, vol. 23, no. 7, pp. 2031–2038, 2013.

[81] K. Swersky, J. Snoek, and R. P. Adams, “Multi-Task Bayesian Optimization,” Advances in Neural Information Processing Systems, vol. 26, pp. 2004–2012, 2013.

[82] K. Swersky, J. Snoek, and R. P. Adams, “Freeze-thaw bayesian optimization,” arXiv preprint arXiv:1406.3896, 2014.

[83] S. Theodoridis, Machine learning: a Bayesian and optimization perspective. Academic press, 2015.

[84] L. Tian, H. Wang, Y. Zhou, and C. Peng, “Video big data in smart city: Background construction and optimization for surveillance video processing,” Future Generation Computer Systems, vol. 86, pp. 1371–1382, 2018.

[85] M. E. Tipping, “Bayesian Inference: An Introduction to Principles and Practice in Machine Learning,” Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003,

Tübingen, Germany, August 4 - 16, 2003, Revised Lectures. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 41–62, 2004. [Online]. Available: https://doi.org/10.1007/978-3-540-28650-9_3

[86] H. M. Torun, M. Swaminathan, A. K. Davis, and M. L. F. Bellaredj, "A global Bayesian optimization algorithm and its application to integrated system design," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 4, pp. 792–802, 2018.

[87] H. Tyagi and V. Cevher, "Active Learning of Multi-Index Function Models.," in NIPS, 2012, pp. 1475–1483.

[88] A. H. Victoria and G. Maragatham, "Automatic tuning of hyperparameters using Bayesian optimization," Evolving Systems, vol. 12, pp. 217–223, 2021.

[89] Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. De Freitas, "Bayesian Optimization in High Dimensions via Random Embeddings.," in IJCAI, 2013, pp. 1778–1784.

[90] L. Wang, M. Feng, B. Zhou, B. Xiang, and S. Mahadevan, "Efficient hyper-parameter optimization for NLP applications," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2112–2117.

[91] Z. Wang, C. Li, S. Jegelka, and P. Kohli, "Batched high-dimensional Bayesian optimization via structural kernel learning," in International Conference on Machine Learning, 2017, pp. 3656–3664.

[92] Z. Wang, C. Gehring, P. Kohli, and S. Jegelka, "Batched large-scale bayesian optimization in high-dimensional spaces," in International Conference on Artificial Intelligence and Statistics, 2018, pp. 745–754.

[93] Y. Wang et al., "Prediction of tubular solar still performance by machine learning integrated with Bayesian optimization algorithm," Applied Thermal Engineering, vol. 184, p. 116233, 2021.

[94] C. White, W. Neiswanger, and Y. Savani, "Bananas: Bayesian optimization with neural architectures for neural architecture search," arXiv preprint arXiv:1910.11858, 2019.

[95] J. Wu, M. Poloczek, A. G. Wilson, and P. I. Frazier, "Bayesian optimization with gradients," arXiv preprint arXiv:1703.04389, 2017.

[96] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," Journal of Electronic Science and Technology, vol. 17, no. 1, pp. 26–40, 2019.

[97] G. Xu, M. Xing, L. Zhang, Y. Liu, and Y. Li, "Bayesian inverse synthetic aperture radar imaging," IEEE Geoscience and Remote Sensing Letters, vol. 8, no. 6, pp. 1150–1154, 2011.

[98] B. Xue et al., "Bayesian transformer language models for speech recognition," in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 7378–7382.

[99] Z. B. Zabinsky, G. Pedrielli, and H. Huang, "A Framework for Multi-fidelity Modeling in Global Optimization Approaches," Cham, 2019, pp. 335–346.

[100] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee, "Improving object detection with deep convolutional networks via bayesian optimization and structured prediction," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 249–258.

[101] T. Zhang, Q. Zhao, K. Shin, and Y. Nakamoto, "Bayesian-optimization-based peak searching algorithm for clustering in wireless sensor networks," Journal of Sensor and Actuator Networks, vol. 7, no. 1, p. 2, 2018.

[102] M. Zhang, H. Li, and S. Su, "High dimensional Bayesian optimization via supervised dimension reduction", in International Joint Conference on Artificial Intelligence 2019.

[103] S. Zhang, W. Lyu, F. Yang, C. Yan, D. Zhou, and X. Zeng, "Bayesian optimization approach for analog circuit synthesis using neural network," in 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2019, pp. 1463–1468.