High Dimensional Bayesian Optimization via Restricted Projection Pursuit Models

Chun-Liang Li **Kirthevasan Kandasamy** **Barnabás Póczos** **Jeff Schneider**
{chunlial, kandasamy, bapoczos, schneide}@cs.cmu.edu
Carnegie Mellon University

Abstract

Bayesian Optimization (BO) is commonly used to optimize blackbox objective functions which are expensive to evaluate. A common approach is based on using Gaussian Process (GP) to model the objective function. Applying GP to higher dimensional settings is generally difficult due to the curse of dimensionality for nonparametric regression. Existing works makes strong assumptions such as the function is low-dimensional embedding (Wang et al., 2013) or is axis-aligned additive (Kandasamy et al., 2015). In this paper, we generalize the existing assumption to a projected-additive assumption. Our generalization provides the benefits of i) greatly increasing the space of functions that can be modeled by our approach, which covers the previous works (Wang et al., 2013; Kandasamy et al., 2015) as special cases, and ii) efficiently handling the learning in a larger model space. We prove that the regret for projected-additive functions has only linear dependence on the number of dimensions in this general setting. Directly using projected-additive GP (Gilboa et al., 2013) to BO results in a non-box constraint, which is not easy to optimize. We tackle this problem by proposing a restricted-projection-pursuit GP for BO. We conduct experiments on synthetic examples and scientific and hyper-parameter tuning tasks in many cases. Our method outperforms existing approaches even when the function does not meet the projected additive assumption. Last, we study the validity of the additive and projected-additive assumption in practice.

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51.

1 Introduction

Many applications in machine learning, science and engineering can be treated as zeroth-order optimization of a smooth function which is not analytically available and is usually expensive to evaluate. By querying the underlying objective function f, the goal is to find those points that have maximal function values.

Bayesian Optimization is a popular method to solve zeroth-order optimization. A commonly-used approach is modeling the unknown objective function f with a Gaussian Process (GP) (Mockus and Mockus, 1991). At time t, BO estimates the posterior GP from queried points and constructs an acquisition function φ_t using upper confidence bound, Thompson sampling, or other heuristics (Brochu et al., 2010). Then BO maximizes φ_t to determine $\mathbf{x}_t$, the next point to be queried. Acquisition functions such as upper confidence bound also apply to the bandit setting and GP assumptions. Gaussian process bandits and Bayesian optimization (GPB/ BO) have been successfully applied in many applications (Brochu et al., 2010). Besides Gaussian Process, other models including random forests (Hutter et al., 2011) and deep neural network (Snoek et al., 2015) are also applied to BO. However, neither of them has the regret analysis with asymptotic results.

Although there are many low dimensional applications (Wang et al., 2013) where existing algorithms work well (typically dimension< 10), there are also important high dimensional applications (Yamins et al., 2013; Gonzalez et al., 2014) for which satisfying high dimensional optimization methods haven not been developed yet. High-dimensional BO is generally nontrivial on both the statistical and computational sides. On the statistical side, GPB/ BO is exponentially difficult in high dimensions with regard to query complexity (Srinivas et al., 2010). In addition, there is the computational cost of optimizing φ_t. Commonly used algorithms require computational time that is exponential in dimension. BO literature typically assumes

that the cost for maximising φ_t is negligible in comparison to the cost of evaluation of f. While this might be true in some cases, in several cases we are constrained in time to optimize φ_t, such as the parameter tunning in machine learning applications (Snoek et al., 2012).

To address the difficulty of high dimensionality, the *low-dimensional assumption* (Chen et al., 2012; Wang et al., 2013; Djolonga et al., 2013) assumes the function only varies within a low-dimensional subspace. On the other hand, Kandasamy et al. (2015) make the *additive assumption* by assuming the function f is a sum of functions of small, disjoint groups of dimensions. The sum is high-dimensional while each term of the sum is low-dimensional. The former approach limits a high-dimensional function in a low dimensional embedding, while the latter is restricted to an axis-aligned representation. We seek to get the best of both of these methods without their corresponding drawbacks.

In this paper, we make the following contributions. First, we generalize the existing two assumptions to a projected-additive assumption (Hastie and Tibshirani, 1990) to handle a broader class of functions. The additive assumption and the low-dimensional assumption are special cases under the projected-additive assumption. We show the regret has only linear dependence on the dimension D when f is under the project-additive assumption (Section 4). Second, although Gilboa et al. (2013) study GPs with a projected-additive approximation, it is non-trivial how to directly apply Gilboa et al. (2013) to BO in practice. After projection, the size of the distorted function domain could be exponentially increased which causes difficulty for optimizing the acquisition functions. We propose the restricted-projection-pursuit algorithm to solve this difficulty (Section 4.1) with good empirical performance on both synthetic and real-world data (Section 5). Last, we also study the case when f does *not* meet these assumptions but still work in practice with theoretical explanations (Section 6).

2 Preliminaries

We aim to maximize a function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X}$ is a *rectangular* region in $\mathbb{R}^D$. Without loss of generality, we assume $\mathcal{X} = [0, 1]^D$. We also assume that we are not allowed to know the exact function formulation but only know the function evaluation by querying at $\mathbf{x} \in \mathcal{X}$ and obtain a noisy observation $y = f(\mathbf{x}) + \epsilon$, where ϵ is Gaussian white noise.

We denote the optimal point as $\mathbf{x}_\star = \text{argmax}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$. At time t, we query point $\mathbf{x}_t$ and incur the *instantaneous regret* $r_t = f(\mathbf{x}_\star) - f(\mathbf{x}_t)$. In the bandit setting, we are interested in minimizing the *cumulative regret* $R_T = \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} (f(\mathbf{x}_\star) - f(\mathbf{x}_t))$, where T is the total number of iterations. In the optimization setting, we are interested in minimizing the *simple regret* $S_T = \min_{t\le T} r_t = f(\mathbf{x}_\star) - \max_t f(\mathbf{x}_t)$. Note that $S_T \le \frac{1}{T} R_T$, so any algorithm with asymptotic sublinear cumulative regret also has vanishing simple regret.

In high-dimensional optimization problems, usually we assume some smoothness conditions on f to make the problem tractable. We assume f is sampled from a zero mean Gaussian Process (Rasmussen and Williams, 2006) with a covariance kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\epsilon \sim \mathcal{N}(0, \eta^2)$. Two commonly used kernels are the squared exponential (SE) and Matérn kernels (Rasmussen and Williams, 2006).

At each time t, we construct the acquisition function φ_t and maximize it to determine which point $\mathbf{x}_t$ to query. A common acquisition function used in bandit problems is the Upper Confidence Bound (UCB) (Cox and John, 1997). In this work, we focus on using the Gaussian Process Upper Confidence Bound (**GP-UCB**) (Srinivas et al., 2010) to construct φ_t. That is, $\varphi_t(\mathbf{x}) = \mu_{t-1}(\mathbf{x}) + \beta_t^{1/2}\sigma_{t-1}(\mathbf{x})$, and $\mathbf{x}_t = \text{argmax}_{\mathbf{x}} \varphi_t(\mathbf{x})$. Here, μ_{t-1} is the posterior GP mean after $t-1$ points, σ_{t-1} is the posterior standard deviation and $\beta_t^{1/2}$ is a coefficient to negotiate the tradeoff between exploration and exploitation. We follow Brochu et al. (2010) to use the Dividing Rectangles (DiRect) algorithm (Jones et al., 1993) to optimize the acquisition function. The time complexity of DiRect is $O(\epsilon^{-D})$ if we want to achieve ϵ accuracy, which makes high dimensional BO challenging given the limited computational time in practice.

3 Additive Gaussian Process Models

To deal with high dimensional problems in the GP framework, Duvenaud et al. (2011) use an additive kernel. Kandasamy et al. (2015) use a group-additive kernel for BO by assuming independence between groups of dimensions. They demonstrate that it achieves significant performance gains in high dimensions. In this section, we briefly review the group-additive model of Kandasamy et al. (2015) and abuse the word "additive" to mean "group-additive" for simplicity.

Assume the function f can be decomposed into the following group-additive form,

$$f(\mathbf{x}) = f^{(1)}(\mathbf{x}^{(1)}) + \ldots + f^{(M)}(\mathbf{x}^{(M)}), \qquad (1)$$

and the mean and kernel functions can be decomposed as well:

$$\mu(x) = \mu^{(1)}(\mathbf{x}^{(1)}) + \ldots + \mu^{(M)}(\mathbf{x}^{(M)}) \qquad (2)$$
$$\kappa(x, x') = \kappa^{(1)}(\mathbf{x}^{(1)}, \mathbf{x}^{(1)\prime}) + \ldots + \kappa^{(M)}(\mathbf{x}^{(M)}, \mathbf{x}^{(M)\prime}),$$

where $\mathbf{x}^{(j)} \in \mathcal{X}^{(j)} = [0,1]^{d_j}$ are disjoint lower dimensional components. For simplicity, assume $d_j = d$ for all j, and thus $D = Md$.

Inference in Additive GPs: By (1), at time t, given the noisy labels $\mathbf{y}_t = \{y_1, \ldots, y_t\}$ at points $\mathbf{X}_t = \{\mathbf{x}_1, \ldots, \mathbf{x}_t\}$ and a query point $\mathbf{x}$, infer the posterior distribution of $f(\mathbf{x})$ by calculating the posterior distributions of $\tilde{f}^{(j)} \equiv f^{(j)}(\mathbf{x}^{(j)})$ individually. For each $\tilde{f}^{(j)}$, the joint distribution can be written as

$$\begin{pmatrix} \tilde{f}^{(j)} \\ \mathbf{y}_t \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \kappa(\mathbf{x}^{(j)}, \mathbf{x}^{(j)}) & \kappa(\mathbf{x}^{(j)}, \mathbf{X}_t^{(j)}) \\ \kappa(\mathbf{X}_t^{(j)}, \mathbf{x}^{(j)}) & \kappa(\mathbf{X}_t, \mathbf{X}_t) + \eta^2 \mathbf{I}_t \end{bmatrix}\right),$$

where the p^{th} element of $\kappa(\mathbf{X}_t^{(j)}, \mathbf{x}^{(j)}) \in \mathbb{R}^t$ is $\kappa(\mathbf{x}_p^{(j)}, \mathbf{x}^{(j)})$ and the $(p,q)^{\text{th}}$ element of $\kappa(\mathbf{X}_t, \mathbf{X}_t) \in \mathbb{R}^{t \times t}$ is $\kappa(\mathbf{x}_p, \mathbf{x}_q)$. Now, the posterior distribution $\mathcal{N}\left(\mu^{(j)}(\mathbf{x}^{(j)}), \sigma^{(j)}(\mathbf{x}^{(j)})\right)$ of $\tilde{f}^{(j)}$ is,

$$\tilde{f}^{(j)} | \mathbf{x}, \mathbf{X}_t, \mathbf{y}_t \sim \mathcal{N}\Big(\kappa(\mathbf{x}^{(j)}, \mathbf{X}_t^{(j)})\Delta^{-1}\mathbf{y}_t, \quad \kappa(\mathbf{x}^{(j)}, \mathbf{x}^{(j)}) - \kappa(\mathbf{x}^{(j)}, \mathbf{X}_t^{(j)})\Delta^{-1}\kappa(\mathbf{X}_t, \mathbf{x}^{(j)})\Big), \tag{3}$$

where $\Delta = \kappa(\mathbf{X}_t, \mathbf{X}_t) + \eta^2 I_t$.

Additive Acquisition Function: Using the inferred posterior distribution of $f^{(j)}(\mathbf{x}^{(j)})$, define the *Additive Gaussian Process Upper Confidence Bound* (**Add-GP-UCB**) to be

$$\begin{aligned} \widetilde{\varphi}_t(\mathbf{x}) = & \quad \mu_{t-1}(\mathbf{x}) + \beta_t^{1/2} \sum_{j=1}^{M} \sigma_{t-1}^{(j)}(\mathbf{x}^{(j)}) \\ = & \quad \sum_{j=1}^{M} \mu_{t-1}^{(j)}(\mathbf{x}^{(j)}) + \beta_t^{1/2} \sigma_{t-1}^{(j)}(\mathbf{x}^{(j)}). \end{aligned} \tag{4}$$

Then the optimization of the acquisition function $\widetilde{\varphi}_t(\mathbf{x})$ can be decomposed as

$$\mathbf{x}_t^{(j)} = \underset{\mathbf{x}^{(j)} \in \mathcal{X}^{(j)}}{\operatorname{argmax}} \mu_{t-1}^{(j)}(\mathbf{x}^{(j)}) + \beta_t^{1/2} \sigma_{t-1}^{(j)}(\mathbf{x}^{(j)}). \tag{5}$$

To achieve ϵ accuracy, the time complexity of optimizing the **Add-GP-UCB** acquisition function is $O(M\epsilon^{-d})$, which is more efficient than $O(\epsilon^{-D})$ of the original UCB algorithm.

The algorithm is shown in Algorithm 1. Note that Kandasamy et al. (2015) learn the decomposition and hyperparameters every N_{cyc} iterations to make the algorithm more efficient.

Algorithm 1 Add-GP-UCB

Input: N_{init}, N_{cyc}, d, M

- $\mathcal{D}_0 \leftarrow$ Uniformly sample N_{init} points from $\mathcal{X}$.
- **for** $t = 1, 2, \ldots$
  1. **if** ($t \mod N_{cyc} = 1$), Learn the Kernel hyper parameters and the decomposition $\{\mathcal{X}_j\}$ by maximising the GP marginal likelihood.
  2. Get $\mathbf{x}_t^{(j)}$ via (5) for all j.
  3. Get y_t via querying f at $\mathbf{x}_t$.
  4. $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$.

4 Projection Pursuit Model

In nonparametric regression, the projected-additive assumption (Hastie and Tibshirani, 1990) is used to generalize the additive assumption. Here, we generalize the projected-additive model used in Gilboa et al. (2013) to a group version as follows,

$$\begin{aligned} f(\mathbf{x}) &= f^{(1)}\left((\mathbf{W}^{(1)})^\top \mathbf{x}\right) + \ldots + f^{(M)}\left((\mathbf{W}^{(M)})^\top \mathbf{x}\right) \\ \mu(\mathbf{x}) &= \mu^{(1)}\left((\mathbf{W}^{(1)})^\top \mathbf{x}\right) + \ldots + \mu^{(M)}\left((\mathbf{W}^{(M)})^\top \mathbf{x}\right) \\ \kappa(\mathbf{x}, \mathbf{x}') &= \kappa^{(1)}\left((\mathbf{W}^{(1)})^\top \mathbf{x}, (\mathbf{W}^{(M)})^\top \mathbf{x}'\right) + \ldots + \\ &\quad \kappa^{(M)}\left((\mathbf{W}^{(M)})^\top \mathbf{x}, (\mathbf{W}^{(M)})^\top \mathbf{x}'\right). \end{aligned} \tag{6}$$

Let $\mathbf{W}$ denote the collection of $\mathbf{W}^{(j)} \in \mathbb{R}^{D \times d}$. We assume $\mathbf{W} \in \mathbb{R}^{D \times D}$, $\mathbf{W}$ is invertible and unknown to us. In the following, we will use "projected-additive" to denote the model in (6).

The proposed group-projected-additive model is a generalization of several previously proposed models. If $\mathbf{W} = \mathbf{I}$, the model is the additive model proposed in Kandasamy et al. (2015). Also, the proposed assumption cover the low-dimensional assumption of Wang et al. (2013) as well.

Algorithm 2 PP-GP-UCB

Input: N_{init}, N_{cyc}, d, M

- $\mathcal{D}_0 \leftarrow$ Uniformly sample N_{init} points from $\mathcal{X}$.
- **for** $t = 1, 2, \ldots$
  1. Learn $\mathbf{W}$ on $\mathcal{D}_{t-1}$ by Gilboa et al. (2013) and get the projected data $\mathbf{Z}_{t-1}$.
  2. Perform Step 1-2 in Algorithm 1 on $\mathbf{Z}_t$ to get $\mathbf{z}_t^{(j)}$ for all j and $\mathbf{x}_t = (\mathbf{W}^{-1})^\top \mathbf{z}_t$.
  3. Get y_t via querying f at $\mathbf{x}_t$.
  4. $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$.

Gilboa et al. (2013) proposed an efficient algorithm called projection pursuit GP regression (**PP-GP**) to approximate $\mathbf{W}$. It partitions $\mathbf{W}$ into several vectors $\mathbf{w}_i$ and learn them individually via the state-space-model using Expectation Maximization. The algorithm utilizes the approximation inference with iteration complexity that is linear time in number of data.

In BO, the number of queries is usually limited, which makes applying **PP-GP** feasible in practice. Observe that the projection pursuit GP regression is equivalent to applying the additive model in Section 3 on the projected data $\mathbf{Z} = \mathbf{XW}$ (Gilboa et al., 2013). Our proposed approach, the **PP-GP-UCB** algorithm for Bayesian Optimization, is described in Algorithm 2, and we also provide a regret bound about its performance in Theorem 2. Due to the space limits, we defer all proofs to the Appendix. First we define the Maximum Information Gain (Srinivas et al., 2010).

Definition 1. *(Maximum information gain) Let* $f \sim \mathcal{GP}(0, \kappa)$ *and* $y_i = f(\mathbf{x}_i) + \epsilon_i$*, where* $\epsilon_i \sim \mathcal{N}(0, \eta^2)$*. Given* $A = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\} \subset \mathcal{X}$*, let* $\mathbf{f}_A = \{f(\mathbf{x}_i)\}$*,* $\mathbf{y}_A = \mathbf{f}_A + \epsilon_A$ *and* I *be the mutual information. The maximum information gain* γ_T *after* T *round is*

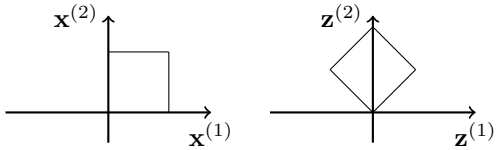
$$\gamma_T = \max_{A \in \mathcal{X}, |A|=T} I(\mathbf{y}_A; \mathbf{f}_A).$$

γ_T characterises the difficulty of the problem and captures the difficulty with dimensionality. For the SE kernel, $\gamma_T \in \tilde{O}((D \log T)^{D+1})$. If it is a sum of SE kernels, Kandasamy et al. (2015) show that it improves to $\tilde{O}(Dd^d (\log T)^{D+1})$.

Theorem 2. *Suppose* $g(\mathbf{z}) = f(\mathbf{x})$*, where* $\mathbf{z} = \mathbf{W}^\top \mathbf{x}$ *and* g *is constructed by sampling* $g^{(j)} \sim \mathcal{GP}(\mathbf{0}, \kappa^{(j)})$ *for* $j = 1, \ldots, M$ *and then summing them as* (6)*. Suppose also that we maximize the acquisition function* $\widetilde{\varphi}_t(\mathbf{z}) = \sum_{j=1}^{M} \mu_{t-1}^{(j)}(\mathbf{z}^{(j)}) + \beta_t^{1/2} \sigma_{t-1}^{(j)}(\mathbf{z}^{(j)})$ *to within* $\tilde{O}(t^{-1/2})$ *accuracy at time step* t *with* $\beta_t = \tilde{O}(d \log t)$*. Under these conditions* **PP-GP-UCB** *attains simple regret* $S_T \in \tilde{O}\left(\sqrt{D\gamma_T \log T / T}\right)$ *with high probability, where* $\gamma_T = O\left(Dd^d (\log T)^{d+1}\right)$ *for SE kernel.*

4.1 Restricted Projection

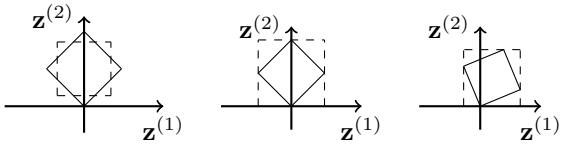
In BO, we maximize a function within a box region $\mathcal{X} = [0, 1]^D$. The box constraints benefit the additive model because we can optimize the acquisition function of each group $\mathbf{x}^{(i)}$ independently under the box constraints $\mathbf{x}^{(j)} \in [0, 1]^{d_j}$ without considering other groups of dimensions.



(a) Original Domain $\mathbf{x} \in [0, 1]^2$ (b) Transformed Domain $(\mathbf{W}^\top)^{-1}\mathbf{z} \in [0, 1]^2$

Figure 1: Two-dimensional example of transformed domains.

When we apply **PP-GP-UCB** to BO, the constraints of the projected data $\mathbf{z}$ is $\left((\mathbf{W}^\top)^{-1}\mathbf{z}\right) \in [0, 1]^D$, which is not a box constraint. We are then not able to optimize the acquisition function by optimizing each group of dimensions $\mathbf{z}^{(i)}$ independently, since all dimensions are correlated. One example is shown in Figure 1. After projection, the range of $\mathbf{z}^{(1)}$ depends on the value of $\mathbf{z}^{(2)}$. Therefore, Algorithm 2 is not applicable without resolving the issue of the transformed domain.



(a) Partial Covered (b) Outer Box (c) Restricted Projection

Figure 2: Two-dimensional example of the different constraints

To deal with the non-box constraint problem, a possible approach is to create a new box constraint to cover the transformed domain. If the new domain only partially cover the transformed domain as shown in Figure 2(a), we may exclude the optimal point, which is less preferable. Therefore, we use the *outer-box domain*, which creates a outer box to fully contain the transformed domain as shown in Figure 2(b).

The problem with the outer-box domain is that we have a larger domain than the original transformed domain, where the size may be exponentially increased in dimensions. Here we focus on the ratio between the sizes of the outer box and the original transformed domain. For instance, if we use grid search to optimize the acquisition function within the outer box in Figure 2(b), then we waste a large number of queries outside of the transformed region.

Observation We could treat the additive model as a special case of projected-additive model with an identity matrix as the projection matrix. Informally speaking, if the projection matrix is similar to the identity matrix, the enhanced region induced by the outer-box domain would be small. However, it is non-trivial to put this constraint into Gilboa et al. (2013).

Inspired by the geometric observation, we propose to use the *restricted projection matrix*, which consists of two steps. We first use the projection matrix $\mathbf{W}$ by Gilboa et al. (2013). We then define the restricted projection matrix as

$$\hat{\mathbf{W}} = (1 - \alpha)\mathbf{W} + \alpha \mathbf{I}, \tag{7}$$

where $\alpha \in [0, 1]$.

If $\alpha = 1$, it is the additive GP regression; if $\alpha = 0$, it is the projection-pursuit GP regression. From the

geometric perspective, we optimize the marginal likelihood while consider the ratio of the size of the outer box to determine α.

$$\begin{aligned} \underset{\alpha}{\text{argmax}} \quad & \log P(\mathbf{y}_t|\mathbf{X}_t;\hat{\mathbf{W}}) \\ \text{subject to} \quad & \frac{|\texttt{outer}(\mathcal{X}_{\hat{\mathbf{W}}})|}{|\mathcal{X}_{\hat{\mathbf{W}}}|} \leq 1+\delta \\ & \hat{\mathbf{W}} = (1-\alpha)\mathbf{W} + \alpha\mathbf{I}, \end{aligned} \tag{8}$$

where $|\mathcal{X}_{\mathbf{W}}| = |\mathbf{W}|$ is the size (volume) of the transformed domain, and $|\texttt{outer}(\mathcal{X}_{\mathbf{W}})| = \prod_{d=1}^{D} \|\mathbf{w}_d\|_1$ is the size of the outer-box domain[1], where $\mathbf{w}_d$ is the d^{th} column of $\mathbf{W}$. By controlling δ, we could restrict the size of searching space when we optimize the acquisition function. We call the constraint $|\texttt{outer}(\mathcal{X}_{\hat{\mathbf{W}}})|/|\mathcal{X}_{\hat{\mathbf{W}}}| \leq 1+\delta$ as *δ-ratio constraint*. The example of the restricted projection is shown in Figure 2(c). Then we only need to optimize one single variable α to control the increased domain, which is simpler than solving $\mathbf{W}$ with δ-ratio constraint directly. In practice, we observe the grid search by checking the constraint could bring us satisfactory empirical performance.

We call the proposed algorithm *Restricted-Projection-Pursuit GP-UCB* (**RPP-GP-UCB**). The pseudo-code of the algorithm is shown in Algorithm 3.

Algorithm 3 RPP-GP-UCB

Input: N_{init}, N_{cyc}, d, M, δ

- $\mathcal{D}_0 \leftarrow$ Uniformly sample N_{init} points from $\mathcal{X}$.
- **for** $t = 1, 2, \ldots$
  1. **if** ($t \mod N_{cyc} = 1$) Learn $\hat{\mathbf{W}}$ with δ-ratio constraint with warm start to make this step more efficient and get the projected data $\mathbf{Z}_{t-1}$.
  2. Perform Step 1-2 in Algorithm 1 on $\mathbf{Z}_t$ with the domain $\texttt{outer}(\hat{\mathbf{W}})$ to get $\mathbf{z}_t^{(j)}$ for all j and $\mathbf{x}_t = (\hat{\mathbf{W}}^{-1})^\top \mathbf{z}_t$.
  3. Get y_t via querying f at $\mathbf{x}_t$.
  4. $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$.

5 Experiment

To make the experiment realistic and demonstrate the efficacy of **RPP-GP-UCB**, we optimize the acquisition function under limited budget with DiRect as used in Brochu et al. (2010). For **RPP-GP-UCB**, we use Gilboa et al. (2013) to find $\mathbf{W}$, and find α in (7) by grid search in the range feasible to the constraint. For grouping the dimensions in **Add-GP-UCB** and **RPP-GP-UCB** , we create the M combinations randomly and choose the one maximizing the marginal likelihood, where M is the number of groups. For all GPB/ BO-based algorithms, we set $N_{init} = 10$, $N_{cyc} = 25$. In all experiments, we use SE kernels and maximize the marginal likelihood $P(\mathbf{y}_t|\mathbf{X}_t;\theta)$ to find the hyperparameters θ. For additive and projected-additive kernels, we use the same hyperparameters for all the kernels. We run each algorithm 50 times for each dataset with different initializations and report the average results.

[1]Note that the original domain is $[0,1]^D$.

5.1 Synthetic Data

First we study the synthetic data to understand the behavior of the proposed algorithm. We generate the objective function f as follows. Let

$$f_{d'}(\mathbf{z}) = \log\big(0.1 \times \texttt{mvnpdf}(\mathbf{z}, \mathbf{v}_1, \sigma^2) + 0.1 \times \texttt{mvnpdf}(\mathbf{z}, \mathbf{v}_2, \sigma^2) + 0.8 \times \texttt{mvnpdf}(\mathbf{z}, \mathbf{v}_3, \sigma^2)\big),$$

where $\sigma^2 = 0.01 d'^{0.1}$ and $\texttt{mvnpdf(x, } \mu\texttt{, } \sigma^2\texttt{)}$ is the probability density at $\mathbf{x}$ of the multivariates Gaussian distribution with mean μ and covariance function $\sigma^2\mathbf{I}$. $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are fixed points in $\mathbb{R}^{d'}$.

We then define $f(\mathbf{x}) = f_{d'}(\mathbf{A}_1^\top \mathbf{x}) + \cdots + f_{d'}(\mathbf{A}_M^\top \mathbf{x})$, where $\mathbf{A}_i \in \mathbb{R}^{D \times d'}$, $\mathbf{x} \in \mathbb{R}^D$ and $D = Md'$. We denote $\mathbf{A} \in \mathbb{R}^{D \times D}$ as the collection of $\mathbf{A}_i$.

We study the cases when $D = 50$ and $D = 100$, with $d' = D/2$ and $\delta = 0.1$. We then set $\mathbf{A} = \mathbf{I} + \mathbf{S}$ and sample each element in $\mathbf{S}$ from $\texttt{Unif}(-1/4, -1/4)$ and $\texttt{Unif}(-1/8, -1/8)$ for $D = 50$ and $D = 100$, respectively to make A invertible. Also, we experiment with **Add-GP-UCB** and **RPP-GP-UCB** for the following values of group dimensionality $d = \{5, 10, 25\}$. Finally, we report the average simple regret of **Add-GP-UCB** (ADD-d) and **RPP-GP-UCB** (RPP-d) with standard errors in Table 1 and 2. More experiment results for other algorithms (Wang et al., 2013; Hutter et al., 2011) can be found in Appendix.

From Table 1, for $D = 50$, neither **Add-GP-UCB** nor **RPP-GP-UCB** are statistically expressive when we use $d = 5$. When we increase d to 10 and 25, both **Add-GP-UCB** and **RPP-GP-UCB** outperform **GP-UCB** . Also, we could notice that $d = 10$ outperforms $d = 25$. Although we set $d = 25$, which is equal to d', optimizing in 25 dimensions is more difficult than optimizing in 10 dimensions. The imperfect optimization of φ_t and the difficulty of estimating functions in higher dimensions degrade the performance.

In Table 2, we have similar observations for $D = 100$. It is worth noticing that optimizing acquisition function gets more difficult under higher dimensional setting. When $D = 100$, both **Add-GP-UCB** and **RPP-GP-UCB** perform better than **GP-UCB** for all values of d.

	GP-UCB	ADD-5	RPP-5	ADD-10	RPP-10	ADD-25	RPP-25
$T = 500$	103.9 ± 1.9	180.3 ± 10.1	151.1 ± 9.7	35.4 ± 2.2	$\mathbf{30.2 \pm 1.7}$	164.5 ± 9.4	152.4 ± 6.5
$T = 1000$	91.4 ± 1.3	153.4 ± 7.1	136.8 ± 8.0	31.3 ± 2.1	$\mathbf{25.1 \pm 1.9}$	77.8 ± 3.4	75.7 ± 2.4

Table 1: The simple regret results of the synthetic data with $D = 50$.

	GP-UCB	ADD-5	RPP-5	ADD-10	RPP-10	ADD-25	RPP-25
$T = 500$	114.1 ± 1.0	168.0 ± 8.1	150.6 ± 12.2	109.8 ± 3.6	$\mathbf{98.9 \pm 2.6}$	175.8 ± 3.7	159.0 ± 2.6
$T = 1000$	106.6 ± 0.9	92.6 ± 3.0	89.5 ± 5.0	41.5 ± 1.4	$\mathbf{34.5 \pm 1.1}$	85.8 ± 1.7	85.1 ± 1.5

Table 2: The simple regret results of the synthetic data with $D = 100$.

To study whether **RPP-GP-UCB** is significantly better than **Add-GP-UCB** , we conduct the t-test at 95% confidence level. The results are given in Tables 3 and 4. Note that when $\mathbf{A}$ is diagonal, f is additive. Since $\mathbf{A}$ is similar to a diagonal matrix but with small perturbations, we could still approximate the function well. Therefore, it is not surprising that the performance difference is not significant when we use large d (e.g. $d = 25$). However, in terms of the average performance, **RPP-GP-UCB** still brings us the performance gain under this situation.

	$d = 5$	$d = 10$	$d = 25$
$T = 500$	win	win	tie
$T = 1000$	win	win	tie

Table 3: **RPP-GP-UCB** versus **Add-GP-UCB** based on t-test at 95% confidence level when $D = 50$. "win" means that **RPP-GP-UCB** is better than **Add-GP-UCB** at the 95% significance level.

	$d = 5$	$d = 10$	$d = 25$
$T = 500$	tie	win	win
$T = 1000$	tie	win	tie

Table 4: **RPP-GP-UCB** versus **Add-GP-UCB** based on t-test at 95% confidence level when $D = 100$.

Without any restriction on $\mathbf{W}$ ($\delta = \infty$), it is equivalent to directly applying Gilboa et al. (2013) on BO. However, the average simple regret is more than 500 when $D = 50$ and $D = 100$. We address the bad performance to two reasons. First, without any restriction, the size of the outer box is much larger, which causes the algorithm query many points out of the original domain. Second, we observe that $\mathbf{W}$ tends to overfit the queried points due to the insufficient queries[2].

5.1.1 Study of δ

Finally we sudy the trade-offs in choosing δ. In Table 5, lager δ results in better performance because it is more statistically expressive. If we are allowed to use more DiRect evaluations, we could expect larger δ could further improve the performance. When d is large enough, such as $d = 25$, even the small δ could be statistically expressive. So using larger δ is less efficient because the algorithm searches a larger domain.

[2]It is also difficult to use any standard approach, such as cross-validation, to avoid overfitting with limited queries.

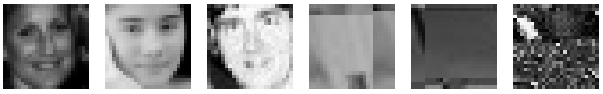


Figure 3: The sample faces and non faces of VJ data.

Note that **RPP-GP-UCB** is still better than **Add-GP-UCB** even though we use smaller δ. It demonstrates the validity of the proposed algorithm.

	$d = 5$	$d = 10$	$d = 25$
$\delta = 0$ (ADD)	53.4 ± 7.1	31.3 ± 2.1	77.8 ± 3.4
$\delta = 0.05$	144.8 ± 9.7	27.9 ± 1.8	$\mathbf{73.3 \pm 3.6}$
$\delta = 0.1$	$\mathbf{136.8 \pm 9.7}$	$\mathbf{25.1 \pm 1.9}$	75.7 ± 2.4

Table 5: The simple regret of **RPP-GP-UCB** when $D = 50$ and $T = 1000$

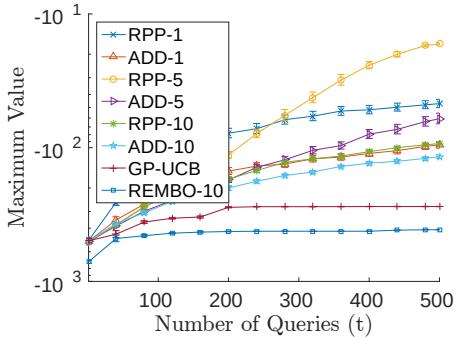
5.2 Real-world Data

Here we consider two real-world tasks, an astrophysical simulation using data from the Sloan Digital Sky Survey (SDSS) (Tegmark et al, 2006) and face detection using the Viola & Jones Cascade classifier (VJ) (Viola and Jones, 2001). The software of SDSS computes the likelihood of a simulation based astrophysical model with cosmological parameters and spectrum data. Higher likelihood could help cosmologists understand the links between these parameters and the power spectra. Therefore, we wish to find the parameters to maximize the likelihood value. The software uses 9 parameters. To emulate the real-world setting, we augment it to 40 dimensions with dummy variables and perturb the parameters with small correlations by $\mathbf{I}_{40\times 40} + \mathbf{S}$, where $\mathbf{S}$ is sampled from $\texttt{Unif}(-0.1, 0.1)$. On the other hand, VJ is a machine learning parameter tunning task for detecting whether an image contains a face. The sampled faces are shown in Figure 3. VJ uses K weak classifiers with K thresholds. If the scores from K classifiers of images pass K thresholds, it is classified as positive. We use the implementation in OpenCV (Bradski and Kaehler, 2008) with $K = 22$ parameters without augmentation.

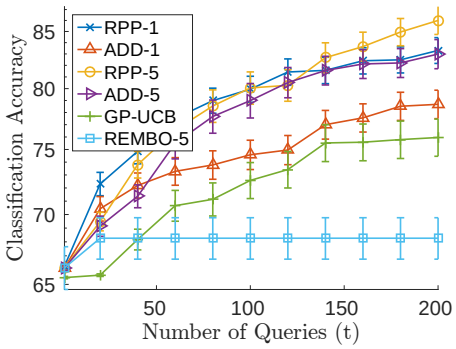
Since the SDSS simulation only takes 3-5 seconds, we use 500 DiRect evaluations for SDSS. We use 1000 evaluations for VJ since the classification takes 30-40 seconds. We also compare with REMBO (Wang et al., 2013) with low-dimensional assumption. For REMBO, we use $\lfloor D/4 \rfloor$ low dimensions. For **RPP-GP-UCB**, we

set $\delta = 0.05$. The average results with standard errors are shown in Figure 4, where `ADD/RPP-d` means each group has d dimensions and `REMBO-d` assumes the number of low dimensions is d.

In practice, the correlation between different dimensions may be larger than the synthetic data used in Section 5.1, then the additive model may not have satisfactory performance by loosing some more information. According to Figure 4, we could observe **RPP-GP-UCB** with small δ significantly outperforms **Add-GP-UCB** when they use the same d. Even when we use $d = 1$ for **RPP-GP-UCB** , it could result in competitive or better performance than **Add-GP-UCB** with larger d, which justifies **RPP-GP-UCB** is more statistically expressive than other models. On the other hand, REMBO does not have satisfactory performance. For SDSS, even though $D/4$ is larger than the number of the underlying low dimensions (9 dimensions). We address the bad performance into the imperfect optimization for acquisition functions, and the empirical choice of the search domain used in Wang et al. (2013) is too small to include the optimal point. For VJ, since the underlying low dimensional assumption does not hold, optimizing $D/4$ dimensions is not enough to get satisfactory performance.



(a) SDSS Astrophysical Experiment



(b) VJ Face Detection.

Figure 4: The results on the real-world data (The higher values are better).

6 Discussion

The additive assumption is usually non-realistic. However, Kandasamy et al. (2015) show that it still works well in practice without a theoretical explanation. In this section, we aim to provide some theoretical insights to understand when the projected-additive assumption works even though the objective function does not meet the additive assumption. For simplicity, we assume the number of dimension of each lower dimensional group $d_j = 1$ for all j, but one can easily generalize the following results to $d_j > 1$.

Assumption 3. *Let $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} = [0,1]^D$. Define $\tilde{f}(\mathbf{z}) = f(\sqrt{D}\mathbf{z})$. The domain $\mathcal{Z}$ of $\tilde{f}$ is $[0, 1/\sqrt{D}]^D$, and $\|\mathbf{z}\| = \epsilon$, where $\epsilon < 1$. Suppose that $\tilde{f}$ has bounded second-order and third-order derivatives.*

Proposition 4. *Under Assumption 3, there exists an additive approximation g (1) modeling the first order derivative such that $\max_{\mathbf{z}} |\tilde{f}(\mathbf{z}) - g(\mathbf{z})| = O(\epsilon^2)$.*

Proposition 5. *Under Assumption 3, there exists a projected-additive approximation h (6) modeling the first and second order derivatives such that $|\tilde{f}(\mathbf{z}) - h(\mathbf{z})| = O(\epsilon^3)$.*

Even though $f(\mathbf{x})$ is not additive, if $f(\mathbf{x})$ is not too wiggly and has bounded high-order derivatives, which is a common assumption in nonparametric regression, then the additive model could be treated as a first-order approximation and the projected-additive model as a second-order approximation.

As we show in Section 5.1, since $\mathbf{A}$ is similar to an identity matrix, the correlation between dimensions is small. Therefore, by only utilizing the first-order approximation (**Add-GP-UCB**) could still result in satisfactory performance. However, the correlation between different dimensions may be larger than the used synthetic data, then we could see **RPP-GP-UCB** significantly outperforms **Add-GP-UCB** in Figure 4 by modeling the second-order information.

Next, we show that the difference between the suboptimality that arises through the approximation of the function and the global optimal is bounded.

Assumption 6. *The posterior mean function is defined as $\mu_t(\mathbf{x}) = \kappa(\mathbf{x}, \mathbf{X}_t) (\kappa(\mathbf{X}_t, \mathbf{X}_t) + \eta \mathbf{I})^{-1} \mathbf{y}_t$ at time t. Let $\tilde{\mathbf{y}}_t = \{\tilde{y}_1, \ldots, \tilde{y}_t\}$, where $|y_i - \tilde{y}_i| \leq \epsilon$. Suppose that $|\mu_t(\mathbf{x}) - \tilde{\mu}_t(\mathbf{x})| < C\epsilon, \forall \mathbf{x}$ and t, where $\tilde{\mu}_t(\mathbf{x}) = \kappa(\mathbf{x}, \mathbf{X}_t) (\kappa(\mathbf{X}_t, \mathbf{X}_t) + \eta \mathbf{I})^{-1} \tilde{\mathbf{y}}_t$ and C is a constant.*

Theorem 7. *Assume g is constructed by sampling from $g \sim \mathcal{GP}(\mathbf{0}, \kappa(\cdot))$, where κ is a SE kernel. Define $\beta_t = \tilde{O}(d \log t)$. Running **GP-UCB** with β_t and the biased samples $(\mathbf{x}_i, \tilde{y}_i)$ instead of noisy samples $(\mathbf{x}_i, y_i)$,*

where $\tilde{y}_i = f(\mathbf{x}_i) + \epsilon_i$ *and* $y_i = g(\mathbf{x}_i) + \epsilon_i$. *Assuming* $|f(\mathbf{x}) - g(\mathbf{x})| \leq \epsilon$, *where* ϵ *is a constant, we then obtain the simple regret bound* $\tilde{O}(\frac{\sqrt{d\gamma_T}}{\sqrt{T}} + \epsilon)$ *with high probability.*

Corollary 8. *The simple regret of* **GP-UCB** *on function* f *can be bounded as* $\tilde{O}(\frac{\sqrt{d\gamma_T}}{\sqrt{T}} + \epsilon)$.

Theorem 7 and Corollary 8 are intuitive by considering the case that the bias is exactly ϵ for every $\mathbf{x}$, that is $g(\mathbf{x}) - f(\mathbf{x}) = \epsilon$. Then the difference between simple regrets with original noisy samples and the biased sample should be $0 < \epsilon$. Therefore, if the approximation error is small, **Add-GP-UCB** and **PP-GP-UCB** can provide a satisfactory performance in practice, since the imperfect optimization of φ_t in **GP-UCB** may cause larger error than ϵ.

6.1 Validity of RPP-GP-UCB

We first consider the case when $\alpha = 0$, which implies $\mathbf{W}$ satisfies the δ-ratio constraint. Assume for simplicity that the optimal point $\mathbf{z}_*$ is not on the boundary of the domain. Then the Hessian matrix $\mathbf{H}_*$ of the $\tilde{f}$ at $\mathbf{z}_*$ is negative definite. If $\mathbf{H}_*$ further has small enough off-diagonal elements such that there exists a matrix $\mathbf{Q}$ satisfying δ-ratio constraint and $\mathbf{H}_* = -\mathbf{Q}\mathbf{Q}^\top$. In turn, the following corollary holds.

Proposition 9. *If* $\tilde{f}$ *satisfies Assumption 3, and the Hessian matrix of* $\tilde{f}(\mathbf{z}_*)$ *can be decomposed as* $-\mathbf{Q}\mathbf{Q}^\top$, *where* $\mathbf{Q}$ *satisfies the* δ*-ratio constraint, there exists a projected-additive approximation* h *such that* $|\tilde{f}(\mathbf{z}) - h(\mathbf{z})| = O(\epsilon^3)$, *where* $\epsilon = \|\mathbf{z} - \mathbf{z}_*\|$, *and* $\mathbf{z}_*$ *is the optimal point of* $\tilde{f}$.

The mild-assumption in Corollary 9 is valid when the correlation between different dimensions is small. A special case is the additive function, where the correlation between different dimensions is zero.

If $\alpha > 0$, which means $\mathbf{W}$ does not satisfy the δ-ratio constraint, which is possible when δ is small. From the approximation perspectives as shown in Proposition 4 and Proposition 5, optimizing (8) is seeking for an intermediate approximation between the first-order and the second-order approximation. The result could be justified by Section 5.1.1 empirically. Even if we use small δ, we could still get performance gain.

6.2 Practical Concern of δ

If we set δ to be large, then we still need to search in a large domain. For example, the worst case is $\alpha = 1$ if δ is large enough. If we do not have enough computational budget to optimize the acquisition function, then we can only use small δ in practice. Note that δ determines how expressive our model is and therefore characterises a bias variance tradeoff. If we set δ smaller, we increase the bias of the estimated $\mathbf{W}$. However, note that the optimization of finding $\mathbf{W}$ (Gilboa et al., 2013) is non-convex, and we are only able to find the local optimum. When D is large, it is impossible to reconstruct $\mathbf{W} \in \mathbb{R}^{D\times D}$ using a limited number of queries without any assumptions. Therefore, (7) plays a role as regularization to reduce the variance of estimating the transform matrix $\mathbf{W}$ given δ.

Intuitively, the **RPP-GP-UCB** is inspired from the geometrical interpretation to reduce the search space; algorithmically, **RPP-GP-UCB** seeks for a intermediates between the first-order and second-order approximation and balanced the trade-off between the increased search domain and the space of the function representation; empirically, we demonstrate **RPP-GP-UCB** outperforms existing approaches even though we use small δ.

7 Conclusion

In this paper, we generalize the existing additive model to projected additive model with regret analysis. We propose **RPP-GP-UCB** to solve the difficulty of the increased function domain, and demonstrate the efficacy and validity of the proposed **RPP-GP-UCB** from both theoretical and empirical sides. We provide a theoretical study to justify the validity of the additive and projected-additive model, which could be treated as first-order and second-order approximation, respectively. Even the function is not projected-additive, the difference between achieved suboptimal and global optimal is bounded under certain conditions.

The empirical study on synthetic and real-world data demonstrates the proposed **RPP-GP-UCB** outperforms the additive model (Kandasamy et al., 2015) and low-dimensional assumption model (Wang et al., 2013). In practice, we observe that even if the small δ, which only model partial correlation between different dimensions, could benefit a lot. On the other hand, if the low dimensional assumption holds, Wang et al. (2013) may have a promising result. However, it is possible that it still has dozens of reduced dimensions. As the empirical results suggest, **RPP-GP-UCB** could still benefit the reduced low dimensional problem.

Our work provides a more expressive framework for BO in high dimensions. Although we do not focus on perfect learning for **RPP-GP**, it already gives us a significant performance gain. However, we observe that the better optimization for **RPP-GP** results in better BO performance. Therefore, it worth studying the better learning algorithm for **RPP-GP** to further boost the performance.

References

Bradski, G. and Kaehler, A. (2008). *Learning OpenCV*. O'Reilly Media Inc.

Brochu, E., Cora, V. M., and de Freitas, N. (2010). A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *CoRR*.

Chen, B., Castro, R., and Krause, A. (2012). Joint Optimization and Variable Selection of High-dimensional Gaussian Processes. In *Proceedings of the International Conference on Machine Learning*.

Cox, D. D. and John, S. (1997). Sdo: A statistical method for global optimization. In *Multidisciplinary Design Optimization: State-of-the-Art*.

Djolonga, J., Krause, A., and Cevher, V. (2013). High-Dimensional Gaussian Process Bandits. In *Advances in Neural Information Processing Systems*.

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. (2011). Additive gaussian processes. In *Advances in Neural Information Processing Systems*.

Gilboa, E., Saatci, Y., Cunningham, J. P., and Gilboa, E. (2013). Scaling multidimensional gaussian processes using projected additive approximations. In *Proceedings of the International Conference on Machine Learning*.

Gonzalez, J., Longworth, J., James, D., and Lawrence, N. (2014). Bayesian Optimization for Synthetic Gene Design. In *NIPS Workshop on Bayesian Optimization in Academia and Industry*.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*. London: Chapman & Hall.

Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In *Proceedings of the International Conference on Learning and Intelligent Optimization*.

Jones, D. R., Perttunen, C. D., and Stuckman, B. E. (1993). Lipschitzian Optimization Without the Lipschitz Constant. *Journal of Optimization Theory and Applications*.

Kandasamy, K., Schenider, J., and Póczos, B. (2015). High Dimensional Bayesian Optimisation and Bandits via Additive Models. In *Proceedings of the International Conference on Machine Learning*.

Mockus, J. and Mockus, L. (1991). Bayesian approach to global optimization and application to multiobjective and constrained problems. *Journal of Optimization Theory and Applications*.

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. University Press Group Limited.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M. M. A., Prabhat, and Adams, R. P. (2015). Scalable bayesian optimization using deep neural networks. In *Proceedings of the International Conference on Machine Learning*.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the International Conference on Machine Learning*.

Tegmark et al, M. (2006). Cosmological Constraints from the SDSS Luminous Red Galaxies. *Physical Review*.

Viola, P. A. and Jones, M. J. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. In *Computer Vision and Pattern Recognition*.

Wang, Z., Zoghi, M., Hutter, F., Matheson, D., and de Freitas, N. (2013). Bayesian Optimization in High Dimensions via Random Embeddings. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Yamins, D., Tax, D., and Bergstra, J. S. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the International Conference on Machine Learning*.