

---

# Bayesian Optimization over Hybrid Spaces

---

Aryan Deshwal<sup>1</sup> Syrine Belakaria<sup>1</sup> Janardhan Rao Doppa<sup>1</sup>

## Abstract

We consider the problem of optimizing hybrid structures (mixture of discrete and continuous input variables) via expensive black-box function evaluations. This problem arises in many real-world applications. For example, in materials design optimization via lab experiments, discrete and continuous variables correspond to the presence/absence of primitive elements and their relative concentrations respectively. The key challenge is to accurately model the complex interactions between discrete and continuous variables. In this paper, we propose a novel approach referred as **Hybrid Bayesian Optimization (HyBO)** by utilizing diffusion kernels, which are naturally defined over continuous and discrete variables. We develop a principled approach for constructing diffusion kernels over hybrid spaces by utilizing the additive kernel formulation, which allows additive interactions of all orders in a tractable manner. We theoretically analyze the modeling strength of additive hybrid kernels and prove that it has the *universal approximation* property. Our experiments on synthetic and six diverse real-world benchmarks show that HyBO significantly outperforms the state-of-the-art methods.

## 1. Introduction

A large number of science and engineering applications involve optimizing hybrid spaces (mixture of discrete and continuous input variables) guided by expensive black-box function evaluations. For example, in materials design optimization, discrete variables correspond to the presence/absence of primitive elements and continuous variables correspond to their relative concentrations, and evaluation of each design involves performing an expensive physical lab experiment. A popular and effective framework for optimizing expensive black-box functions is Bayesian optimization (BO) (Shahriari et al., 2016; Frazier, 2018; Greenhill et al., 2020;

Belakaria et al., 2019; Zhou et al., 2020; Belakaria et al., 2020e;c;a;f). The key idea behind BO is to learn a surrogate statistical model and intelligently select the sequence of inputs for evaluation to approximately optimize the unknown objective. Gaussian process (GP) (Rasmussen & Williams, 2006) is the most popular choice for learning statistical models. GPs allow to incorporate domain knowledge about the problem in the form of a kernel over the input space and provide good uncertainty quantification. GPs have been successfully applied for both continuous (Shahriari et al., 2016; Belakaria et al., 2020b;d) and discrete spaces (Oh et al., 2019; Deshwal et al., 2021; Roustant et al., 2020). However, as we discuss in the related work section, there is very limited work on BO methods to optimize hybrid spaces (Hutter et al., 2010; 2011; Bergstra et al., 2011; Daxberger et al., 2020; Ru et al., 2020). Most of them employ non-GP based surrogate models as it is challenging to define a generic kernel over hybrid spaces that can account for complex interactions between variables.

To precisely fill this gap in our knowledge, we propose a novel approach referred as **Hybrid Bayesian Optimization (HyBO)**. HyBO builds GP based surrogate models using diffusion kernels, which are naturally defined over continuous (Kondor & Vert, 2004) and discrete spaces (Kondor & Lafferty, 2002). We develop a principled approach to construct diffusion kernels over hybrid spaces. This approach employs the general formulation of additive Gaussian process kernels (Duvenaud et al., 2011) to define *additive hybrid diffusion* kernels. The key idea is to assign a base kernel for each discrete/continuous variable and construct an overall kernel by summing over all possible orders of interaction between these kernels. This construction procedure has two advantages: 1) Allows to leverage existing kernels for continuous and discrete spaces; and 2) Can automatically identify the strength of different orders of interaction in a data-driven manner for a given application.

A key question about the modeling strength of this hybrid diffusion kernel is whether given sufficient data, can it approximate any black-box function defined over hybrid spaces. This question has been studied in the past in terms of a property called *universality* of a kernel (Steinwart, 2001; Micchelli et al., 2006; Sriperumbudur et al., 2011; Mania et al., 2018). We prove that the proposed hybrid diffusion kernel has universal approximation property by composing

<sup>1</sup>School of EECS, Washington State University, Pullman, USA.  
Correspondence to: Aryan Deshwal <aryan.deshwal@wsu.edu>.

a known result for continuous diffusion kernels with a novel result for discrete diffusion kernels. Our theoretical results have broader significance going beyond the BO literature.

Our experiments on diverse synthetic benchmarks and real-world applications show that HyBO performs significantly better than state-of-the-art methods. We also empirically demonstrate that superiority of HyBO’s performance is due to better surrogate model resulting from the proposed additive hybrid diffusion kernel.

**Contributions.** The key contribution of this paper is the development and evaluation of the HyBO approach to perform BO over hybrid spaces. Specific list includes:

- Development of a principled approach to construct additive diffusion kernels over hybrid spaces for building GP based surrogate statistical models.
- Theoretical analysis to prove that additive hybrid diffusion kernel has the universal approximation property.
- Experiments on synthetic and real-world benchmarks to show that HyBO significantly improves over state-of-the-art methods. The code and data are available on the GitHub repository <https://github.com/aryandeshwal/HyBO>.

## 2. Problem Setup and Hybrid Bayesian Optimization Approach

**Problem Setup.** Let  $\mathcal{X}$  be a hybrid space to be optimized over, where each element  $x \in \mathcal{X}$  is a hybrid structure. Without loss of generality, let each hybrid structure  $x = (x_d \in \mathcal{X}_d, x_c \in \mathcal{X}_c) \in \mathcal{X}$  be represented using  $m$  discrete variables and  $n$  continuous variables, where  $x_d$  and  $x_c$  stands for the discrete and continuous sub-space of  $\mathcal{X}$ . Let each discrete variable  $v_d$  from  $x_d$  take candidate values from a set  $C(v_d)$  and each continuous variable  $v_c$  from  $x_c$  take values from a compact subset of  $\mathbb{R}$ . In parts of the ML literature, a distinction is made between categorical and discrete variables based on their values: *categorical* refers to an unordered set (e.g., different types of optimizers for neural network training) and *discrete* refers to an ordered set (e.g., number of layers in a neural network). We do not make such distinction because our HyBO approach works for both cases. Concretely, by our definition, a categorical variable is also a discrete variable, i.e.,  $C(v_d)$  is just the no. of candidate values for categorical variable  $v_d$ . We are given a space of hybrid structures  $\mathcal{X}$ . We assume an unknown, expensive real-valued objective function  $\mathcal{F} : \mathcal{X} \mapsto \mathbb{R}$ , which can evaluate each hybrid structure  $x$  (also called an experiment) and produces an output  $y = \mathcal{F}(x)$ . For example, in high-entropy alloys optimization application,  $x_d$  corresponds to the presence/absence of metals and  $x_c$  corresponds to their relative concentrations, and  $\mathcal{F}(x)$  corresponds to running a physical lab experiment using additive manufacturing tech-

niques. The main goal is to find a hybrid structure  $x \in \mathcal{X}$  that approximately optimizes  $\mathcal{F}$  by conducting a limited number of evaluations and observing their outcomes.

**Bayesian Optimization Framework.** BO is a very efficient framework to solve global optimization problems using *black-box evaluations of expensive objective functions* (Shahriari et al., 2016). BO algorithms intelligently select the next input for evaluation guided by a learned statistical model to quickly direct the search towards optimal inputs. The three key elements of BO framework are:

- 1) *Statistical model* of the true function  $\mathcal{F}(x)$ . Gaussian Process (GP) (Rasmussen & Williams, 2006) is the most popular choice for statistical model. GPs allows to incorporate domain knowledge by defining an appropriate kernel over the input space and have better uncertainty quantification ability. A GP over a space  $\mathcal{X}$  is a random process from  $\mathcal{X}$  to  $\mathbb{R}$ . It is characterized by a mean function  $\mu : \mathcal{X} \mapsto \mathbb{R}$  and a covariance or kernel function  $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ .
- 2) *Acquisition function* ( $\mathcal{AF}$ ) to score the utility of evaluating a candidate input  $x \in \mathcal{X}$  based on the statistical model  $\mathcal{M}$ . Expected improvement (EI) (Mockus et al., 1978) is a prototypical acquisition function.
- 3) *Optimization procedure* to select the best scoring candidate input for evaluation according to  $\mathcal{AF}$ .

### Algorithm 1 HyBO Approach

**Input:**  $\mathcal{X}$  = Hybrid input space,  $\mathcal{K}(x, x') =$  Kernel over hybrid structures,  $\mathcal{AF}(\mathcal{M}, x) =$  Acquisition function parametrized by model  $\mathcal{M}$  and input  $x$ ,  $\mathcal{F}(x) =$  expensive objective function

**Output:**  $\hat{x}_{best}$ , the best structure

- ```

1: Initialize  $\mathcal{D}_0 \leftarrow$  initial training data; and  $t \leftarrow 0$ 
2: repeat
3:   Learn statistical model:  $\mathcal{M}_t \leftarrow \text{GP-LEARN}(\mathcal{D}_t, \mathcal{K})$ 
4:   Compute the next structure to evaluate:
    $x_{t+1} \leftarrow \arg \max_{x \in \mathcal{X}} \mathcal{AF}(\mathcal{M}_t, x)$ 
5:    $x_c \leftarrow$  Optimize continuous subspace conditioned on assignment to discrete variables  $x_d$ 
6:    $x_d \leftarrow$  Optimize discrete subspace conditioned on assignment to continuous variables  $x_c$ 
7:   Evaluate objective function  $\mathcal{F}(x)$  at  $x_{t+1}$  to get  $y_{t+1}$ 
8:   Aggregate the data:  $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}$ 
9:    $t \leftarrow t + 1$ 
10: until convergence or maximum iterations
11:  $\hat{x}_{best} \leftarrow \arg \max_{x_t \in \mathcal{D}} y_t$ 
12: return the best uncovered hybrid structure  $\hat{x}_{best}$ 
```

**Hybrid Bayesian Optimization Approach.** Our *HyBO* approach is an instantiation of the generic BO framework by instantiating the statistical model and acquisition function optimization procedure for hybrid spaces (see Algorithm 1).

*Statistical model over hybrid structures.* We employ GPs to build statistical models. To accurately model the complex interactions between discrete and continuous variables, we invoke a principled approach to *automatically* construct additive diffusion kernels over hybrid structures by leveraging

diffusion kernels over continuous and discrete spaces.

*Acquisition function optimization.* Suppose  $\mathcal{M}_t$  is the statistical model at iteration  $t$ . Let us assume that  $\mathcal{AF}(\mathcal{M}_t, x)$  is the acquisition function that need to be optimized to select the next hybrid structure  $x_{t+1}$  for function evaluation. We solve this problem using an iterative procedure that performs search over continuous sub-space ( $x_c$ ) and discrete sub-space ( $x_d$ ) alternatively. For searching continuous and discrete sub-spaces, we employ CMA-ES (Hansen, 2016) and hill-climbing with restarts respectively. We observed that *one* iteration of optimizing continuous and discrete sub-spaces gave good results and they were not sensitive to more iterations. All results of HyBO are with one iteration.

### 3. Related Work

The effectiveness of any BO approach over hybrid spaces depends critically on the choice of surrogate model. Prior work explored a variety of surrogate models. SMAC (Hutter et al., 2010) employs random forest, which may suffer from inaccurate uncertainty quantification due to its frequentist estimation. TPE (Bergstra et al., 2011) models each input dimension *independently* by a kernel density estimator, which can be restrictive due to large size of input dimensions and no inter-dependency among models of different input dimensions. MiVaBO (Daxberger et al., 2020) employs a Bayesian linear regressor by defining features that capture the discrete part using BOCS model (Baptista & Poloczek, 2018; Deshwal et al., 2020a), continuous part using random fourier features (Rahimi & Recht, 2007), and pairwise interaction between continuous and discrete features. As the number of parameters increase, it will need a lot of training examples for learning accurate statistical model.

GP based models overcome the drawbacks of all the above methods. (Garrido-Merchán & Hernández-Lobato, 2020) provided a solution for BO over discrete spaces using an input-transformed kernel. A recent work referred as CoCaBO (Ru et al., 2020) employs a sum kernel (summing a Hamming kernel over discrete subspace and a RBF kernel over continuous subspace) to learn GP models and showed good results over SMAC and TPE. Unfortunately, the sum kernel captures limited interactions between discrete and continuous variables. In contrast, our additive hybrid diffusion kernel allows to capture higher-order interactions among hybrid variables and our data-driven approach can automatically learn the strengths of these interactions from training data. HyperBand (HB) (Li et al., 2017) and its model-based variant BOHB (Falkner et al., 2018) are efficient *multi-fidelity methods* for hyper-parameter optimization that build on existing methods to optimize hybrid spaces. Our HyBO approach is complementary to this line of work.

Prior methods perform search over discrete and continuous

subspaces (e.g., gradient descent) to solve the acquisition function optimization problem. SMAC employs a *hand-designed* local search procedure. MiVaBO uses integer program solvers to search discrete subspace. Learning methods to improve the accuracy of search (Deshwal et al., 2020b) are complementary to SMAC, MiVABO, and HyBO. CoCaBO maintains a separate multi-armed bandit for each discrete variable and employs the EXP3 algorithm (Auer et al., 2002) to select their values *independently*. This method does not exploit dependencies among variables, which can be detrimental to accuracy. TPE samples from the learned density estimator to pick the best input for evaluation.

## 4. Diffusion Kernels over Hybrid Structures

We first provide the details of key mathematical and computational tools that are needed to construct hybrid diffusion kernels. Next, we describe the algorithm to automatically construct additive diffusion kernels over hybrid structures. Finally, we present theoretical analysis to show that hybrid diffusion kernels satisfy universal approximation property.

### 4.1. Key Mathematical and Computational Tools

Diffusion kernels (Kondor & Vert, 2004; Lafferty & Lebanon, 2005) are inspired from the diffusion processes occurring in physical systems like heat and gases. The mathematical formulation of these processes naturally lends to kernels over both continuous and discrete spaces(e.g., sequences, trees, and graphs).

**Diffusion kernel over continuous spaces.** The popular radial basis function (RBF) kernel (also known as Gaussian kernel) (Kondor & Vert, 2004) is defined as follows:

$$k(x, x') = \frac{1}{2\pi\sigma^2} e^{-\|x-x'\|^2/2\sigma^2} \quad (4.1)$$

where  $\sigma$  is the length scale hyper-parameter. This is the solution of the below continuous diffusion (heat) equation:

$$\frac{\partial}{\partial t} k_{x_0}(x, t) = \Delta k_{x_0}(x, t) \quad (4.2)$$

where  $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \dots + \frac{\partial^2}{\partial x_D^2}$  is the second-order differential operator known as the *Laplacian operator*, and  $k_{x_0}(x, t) = k(x, x')$  with  $x' = x_0$  and  $t = \sigma^2/2$ .

### 4.2. Diffusion Kernel over discrete spaces

The idea of diffusion kernels for continuous spaces is extended to discrete structures (e.g., sequences, graphs) (Kondor & Lafferty, 2002) by utilizing the spectral properties of a graph representation of the discrete space. A discrete analogue of the Equation 4.2 can be constructed by employing the matrix exponential of a graph and the *graph Laplacian*

operator  $L$  as given below:

$$\frac{\partial}{\partial \beta} e^{\beta L} = L e^{\beta L} \quad (4.3)$$

where  $L$  is the graph Laplacian of a suitable graph representation of the discrete input space and  $\beta$  is a hyper-parameter of the resulting diffusion kernel similar to the length scale parameter  $\sigma$  of the RBF kernel. The solution of Equation 4.3 defines a positive-definite kernel for discrete spaces known as the discrete diffusion kernel.

According to Equation 4.3, one important ingredient required for defining diffusion kernels on discrete spaces is a suitable graph representation for discrete spaces. One such representation was proposed in a recent work (Oh et al., 2019). In this case, the entire discrete space is represented by a combinatorial graph  $G$ . Each node in the vertex set  $V$  of the graph corresponds to one candidate assignment of all the discrete variables. Two nodes are connected by an edge if the Hamming distance between the corresponding assignments for all discrete variables is exactly one. The diffusion kernel over this representation is defined as follows:

$$k(V, V) = \exp(-\beta L(G)) \quad (4.4)$$

$$k(V, V) = \Phi \exp(-\beta \Pi) \Phi^T \quad (4.5)$$

where  $\Phi = [\phi_1, \dots, \phi_{|V|}]$  is the eigenvector matrix and  $\Pi = [\pi_1, \dots, \pi_{|V|}]$  is the eigenvalue matrix, where  $\phi_i$ 's and  $\pi_i$ 's are the eigenvectors and eigenvalues of the graph Laplacian  $L(G)$  respectively. Although this graph representation contains an exponential number of nodes, (Oh et al., 2019) computes the graph Laplacian  $L(G)$  by decomposing it over the Cartesian product ( $\diamond$ ) of  $m$  (number of discrete variables) sub-graphs  $(G_1, G_2 \dots, G_m)$  with each sub-graph  $G_i$  representing one variable individually. This algorithmic approach has time-complexity  $O(\sum_{i=1}^m (C(v_i))^3)$ , where  $C(v_i)$  is the number of candidate values (arity) for the  $i$ th discrete variable. However, this method is computationally expensive, especially, for problems with large-sized arity.

To avoid this computational challenge, we leverage prior observation in (Kondor & Lafferty, 2002) which provides a *closed-form* of the discrete diffusion kernel by exploiting the structure of the above combinatorial graph representation. We explain this observation for binary variables  $\{0, 1\}$ . From its definition in Equation 4.4, the discrete diffusion kernel over single-dimensional input will be:

$$k(x_d, x'_d) = \begin{cases} (1 - e^{-2\beta}) & \text{if } x_d \neq x'_d \\ (1 + e^{-2\beta}) & \text{if } x_d = x'_d \end{cases} \quad (4.6)$$

Since the kernel over  $m > 1$  dimensions is defined using the Kronecker product over  $m$  dimensions, the above expression is easily extended to multiple dimensions setting giving:

$$k(x_d, x'_d) = \prod_{i=1}^m \frac{(1 - e^{-2\beta_i})^{\delta(x_d^i, x'^i_d)}}{(1 + e^{-2\beta_i})^{1 - \delta(x_d^i, x'^i_d)}} \quad (4.7)$$

where  $\delta(x_d^i, x'^i_d) = 0$  if  $x_d^i$  is equal to  $x'^i_d$  and 1 otherwise. The subscript  $d$  denotes that the variables are discrete and the superscript refers to the  $i$ th dimension of the discrete subspace. For general (discrete spaces with arbitrary categories), we follow the same observation (Kondor & Lafferty, 2002) and use the following constant-time expression of the discrete diffusion kernel in our method:

$$k(x_d, x'_d) = \prod_{i=1}^m \left( \frac{1 - e^{-C(v_i)\beta_i}}{1 + (C(v_i) - 1)e^{-C(v_i)\beta_i}} \right)^{\delta(x_d^i, x'^i_d)} \quad (4.8)$$

### 4.3. Diffusion Kernels over Hybrid Spaces

**Unifying view of diffusion kernels.** Our choice of diffusion kernels is motivated by the fact that they can be naturally defined for both discrete and continuous spaces. In fact, there is a nice transition of the diffusion kernel from discrete to continuous space achieved by continuous space limit operation. More generally, both discrete and continuous diffusion kernel can be seen as continuous limit operation on two parameters of random walks: *time* and *space*. For illustration, consider a random walk on an evenly spaced grid where mean time of jump is  $t$  and mean gap between two points is  $s$ . If  $t \rightarrow 0$ , the resulting continuous time and discrete space random walk generates the diffusion kernel on discrete spaces. Additionally, in the limit of the grid spacing  $s$  going to zero, the kernel will approach the continuous diffusion kernel.

**Algorithm to construct hybrid diffusion kernels.** We exploit the general formulation of additive Gaussian process kernels (Duvenaud et al., 2011) to define an *additive hybrid diffusion* kernel over hybrid spaces. The key idea is to assign a base kernel *for each input dimension*  $i \in \{1, 2, \dots, m+n\}$ , where  $m$  and  $n$  stand for the number of discrete and continuous variables in hybrid space  $\mathcal{X}$ ; and construct an overall kernel by summing all possible orders of interactions (upto  $m+n$ ) between these base kernels. In our case, the RBF kernel and the discrete diffusion kernel acts as the base kernel for continuous and discrete input dimensions respectively. The  $p^{\text{th}}$  order of interaction (called  $p^{\text{th}}$  additive kernel) is defined as given below:

$$\mathcal{K}_p = \theta_p^2 \sum_{1 \leq i_1 < i_2 < \dots < i_p \leq m+n} \left( \prod_{d=1}^p k_{i_d}(x_{i_d}, x'_{i_d}) \right)$$

where  $\theta_p$  is a hyper-parameter associated with each additive kernel and  $k_{i_d}$  is the base kernel for the input dimension  $i_d$ .

In words, the  $p$ th additive kernel is a sum of  $\binom{m+n}{p}$  terms, where each term is a product of  $p$  distinct base kernels. Estimation of  $\theta_p$  hyper-parameter from data allows automatic identification of important orders of interaction for a given application. The overall *additive hybrid diffusion kernel*  $\mathcal{K}_{HYB}(x, x')$  over hybrid spaces is defined as the sum of all orders of interactions as given below:

$$\mathcal{K}_{HYB} = \sum_{p=1}^{m+n} \mathcal{K}_p \quad (4.9)$$

$$\mathcal{K}_{HYB} = \sum_{p=1}^{m+n} (\theta_p^2 \sum_{i_1, \dots, i_p} \prod_{d=1}^p k_{i_d}(x_{i_d}, x'_{i_d})) \quad (4.10)$$

It should be noted that the RHS in Equation 4.10 requires computing a sum over exponential number of terms. However, this sum can be computed in polynomial time using Newton-Girard formula for elementary symmetric polynomials (Duvenaud et al., 2011). It is an efficient formula to compute the  $p^{\text{th}}$  additive kernel recursively as given below:

$$\mathcal{K}_p = \theta_p^2 \cdot \left( \frac{1}{p} \sum_{j=1}^p (-1)^{(j-1)} \mathcal{K}_{p-j} S_j \right) \quad (4.11)$$

where  $S_j = \sum_{i=1}^{m+n} k_i^j$  is the  $j$ th power sum of all base kernels  $k_i$  and the base case for the recursion can be taken as 1 (i.e.,  $\mathcal{K}_0 = 1$ ). This recursive algorithm for computing additive hybrid diffusion kernel has the time complexity of  $\mathcal{O}((n+m)^2)$ .

**Data-driven specialization of kernel for a given application.** In real-world applications, the importance of different orders of interaction can vary for optimizing the overall performance of BO approach (i.e., minimizing the number of expensive function evaluations to uncover high-quality hybrid structures). For example, in some applications, we may not require all orders of interactions and only few will suffice. The  $\theta_p$  hyper-parameters in the additive hybrid diffusion kernel formulation allows us to identify the strength/contribution of the  $p$ th order of interaction for a given application in a *data-driven* manner. We can compute these parameters (along with the hyper-parameters for each base kernel) by maximizing the marginal log-likelihood, but we consider a fully Bayesian treatment by defining a prior distribution for each of them. This is important to account for the uncertainty of the hyper-parameters across BO iterations. The acquisition function  $\mathcal{AF}(x)$  is computed by marginalizing the hyper-parameters as given below:

$$\mathcal{AF}(x; \mathcal{D}) = \int \mathcal{AF}(x; D, \Theta) p(\Theta | D) d\Theta \quad (4.12)$$

where  $\Theta$  is a variable representing all the hyperparameters ( $\sigma$  for continuous diffusion kernel,  $\beta$  for discrete diffusion

kernel, and  $\theta$  for strengths of different orders of interaction in hybrid diffusion kernel) and  $\mathcal{D}$  represents the aggregate dataset containing the hybrid structure and function evaluation pairs. The posterior distribution over the hyper-parameters is computed using slice sampling (Neal, 2003).

#### 4.4. Theoretical Analysis

Intuitively, a natural question to ask about the modeling power of a kernel is whether (given enough data) it can approximate (with respect to a suitable metric) any black-box function defined over hybrid spaces. This is a minimum requirement that should guide our choice of kernel in the given problem setting. This question has been studied widely in the form of a key property called *universality* of a kernel (Steinwart, 2001; Micchelli et al., 2006; Sriperumbudur et al., 2011; Mania et al., 2018). In this section, we prove the universality of the *additive hybrid diffusion kernel* by combining the existing result on the universality of RBF (Gaussian) kernel with a novel result proving the universality of discrete diffusion kernels.

**Proposition 1** (Steinwart, 2001; Micchelli et al., 2006) Let  $\mathcal{X}_c$  be a compact and non-empty subset of  $\mathbb{R}^n$ . The RBF kernel in Equation 4.1 is a universal kernel on  $\mathcal{X}_c$ .

A kernel  $k$  defined on an input space  $\mathcal{X}_c$  has a unique correspondence with an associated Reproducing Kernel Hilbert Space (RKHS) of functions  $\mathcal{H}_k$  defined on  $\mathcal{X}_c$  (Steinwart & Christmann, 2008). For compact metric input spaces  $\mathcal{X}_c$ , a kernel  $k$  is called universal if the RKHS  $\mathcal{H}_k$  defined by it is dense in the space of continuous functions  $C(\mathcal{X}_c)$ . (Steinwart, 2001) proved the universality of the RBF (Gaussian) kernel with respect to the uniform norm. (Micchelli et al., 2006) established universality for a larger class of translation invariant kernels. (Sriperumbudur et al., 2011) discussed various notions of universality and connected to the concept of *characteristic kernels*.

**Proposition 2** Let  $\mathcal{X}_d$  be the discrete space  $\{0, 1\}^m$  and a pseudo-boolean function on  $\mathcal{X}_d$  is defined as  $f : \mathcal{X}_d \mapsto \mathbb{R}$ . The discrete diffusion kernel is a universal kernel on  $\mathcal{X}_d$ .

**Proof.** A Reproducing Kernel Hilbert Space  $\mathcal{H}_k$  associated with a kernel  $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$  is defined as:

$$\mathcal{H}_k = cl(span\{k(x, \cdot), \forall x \in \mathcal{X}\}) \quad (4.13)$$

where  $cl$  represents the closure and  $k(x, \cdot)$  is called as the feature map of  $x$  (Steinwart & Christmann, 2008).

In our setting, a kernel  $k$  defined on discrete input space  $\mathcal{X}_d$  is universal if and only if any pseudo-Boolean function  $f$  can be written as a linear combination of functions  $(k(x_{i_d}, \cdot), \forall x_{i_d} \in \mathcal{X}_d)$  in the RKHS  $\mathcal{H}_k$  (Mania et al., 2018; Gretton et al., 2012), i.e.

$$\forall f : \mathcal{X}_d \mapsto \mathbb{R}; \quad \exists a_i \in \mathbb{R}; f = \sum_i a_i k(x_{i_d}, \cdot); \quad (4.14)$$

We prove that this is true by computing the explicit form of functions  $(k(x_{i_d}, \cdot), \forall x_{i_d} \in \mathcal{X}_d)$  existing in the RKHS  $\mathcal{H}_k$  of the discrete diffusion kernel. To see this, we exploit the structure of the combinatorial graph representation of the discrete space discussed in Section 4.1. The discrete diffusion kernel is defined in terms of the eigenvectors  $\phi_i$  and eigenvalues  $\pi_i$  of the graph Laplacian  $L(G)$  as follows:

$$k(x_d, x'_d) = \sum_{i=1}^{2^n} \phi_i[x_d] \exp(-\beta \pi_i) \phi_i[x'_d] \quad (4.15)$$

Since the combinatorial graph  $G$  is generated by the Cartesian product over sub-graphs  $G_i$  (one for each discrete variable), the eigenvectors term  $\phi_i[x_d]$  can be calculated via an explicit formula, i.e.,  $\phi_i[x_d] = -1^{w^T x_d}$ , where  $w$  is a binary vector of size  $n$  (Chung & Graham, 1997) (number of discrete variables).

$$k(x_d, x'_d) = \sum_{i=1}^{2^n} -1^{w^T x_d} \exp(-\beta \pi_i) - 1^{w^T x'_d} \quad (4.16)$$

$$\langle k(x_d, \cdot), k(x'_d, \cdot) \rangle = \sum_{i=1}^{2^n} -1^{w^T x_d} \exp(-\beta \pi_i) - 1^{w^T x'_d} \quad (4.17)$$

where the inner product in LHS follows from the reproducing property (Steinwart & Christmann, 2008) of a kernel  $k$ . Therefore, the functions  $k(x_d, \cdot)$  in the RKHS  $\mathcal{H}_k$  of the discrete diffusion kernel are of the form  $\{-1^{w_j^T x_d}; w_j \in [0, 2^n - 1]\}$ , which is the well-known *Walsh Basis* (Verel et al., 2018) for pseudo-Boolean functions. Therefore, any pseudo-Boolean function  $f$  can be represented by a linear combination of functions in  $\mathcal{H}_k$  since they form a basis.

**Theorem 4.1** Let  $\mathcal{X}_c$  be a compact and non-empty subset of  $\mathbb{R}^n$  and  $\kappa_c$  be RBF kernel on  $\mathcal{X}_c$ . Let  $\mathcal{X}_d$  be the discrete space  $\{0, 1\}^m$  and  $\kappa_d$  be discrete diffusion kernel on  $\mathcal{X}_d$ . The additive hybrid diffusion kernel defined in Eqn 4.10, instantiated with  $\kappa_c$  and  $\kappa_d$  for continuous and discrete spaces respectively, is a universal kernel for the hybrid space  $\mathcal{X}_c \times \mathcal{X}_d$ .

According to Equation 4.9, any  $p$ th order of interaction term in the additive hybrid diffusion kernel is defined as  $(\prod_{d=1}^p k_{i_d}(x_{i_d}, x'_{i_d}))$ . Therefore, if each  $k_{i_d}$  is universal over its corresponding dimension  $X_{i_d}$  (which is true from Propositions 1 and 2), we need to show that the product  $(\prod_{d=1}^p k_{i_d}(x_{i_d}, x'_{i_d}))$  is universal over the union of dimensions  $\mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \cdots \times \mathcal{X}_{i_p}$ . This was proven by Lemma A.5 in (Steinwart et al., 2016). We provide the lemma here for completeness.

**Lemma 4.2** From (Steinwart et al., 2016) Let  $\mathcal{X} \subset \mathbb{R}^m$  be a compact and non-empty subset,  $I, J \subset \{1, \dots, m\}$  be

| Name       | Name in the suite | Dimension    |
|------------|-------------------|--------------|
| Function 1 | f001_i01_d10      | 10 (8d, 2c)  |
| Function 2 | f001_i02_d10      | 10 (8d, 2c)  |
| Function 3 | f001_i01_d20      | 20 (16d, 4c) |
| Function 4 | f001_i02_d20      | 20 (16d, 4c) |

Table 1. Benchmark problems from bbox-mixint suite.

non-empty, and  $k_I$  and  $k_J$  be universal kernels on  $\mathcal{X}_I \times \mathcal{X}_J$ , respectively. Then  $k_I \otimes k_J$  defined by

$$k_I \otimes k_J(x, x') := k_I(x_I, x'_I) \cdot k_J(x_J, x'_J)$$

for all  $x, x' \in \mathcal{X}_I \times \mathcal{X}_J$  is a universal kernel on  $\mathcal{X}_I \times \mathcal{X}_J$ .

Since both continuous and discrete spaces are compact and Lemma 4.2 is true for arbitrary compact spaces, each order of interaction is universal with respect to its corresponding ambient dimension  $\mathcal{X}_{i_1} \times \mathcal{X}_{i_2} \cdots \times \mathcal{X}_{i_p}$ . In particular, it is true for  $m + n$ th order of interaction which is defined over the entire hybrid space  $\mathcal{X}_c \times \mathcal{X}_d$  which proves the theorem.

## 5. Experiments and Results

We first describe our experimental setup. Next, we discuss experimental results along different dimensions.

### 5.1. Benchmark Domains

**Synthetic benchmark suite.** `bbox-mixint` is a challenging mixed-integer blackbox optimization benchmark suite (Tušar et al., 2019) that contains problems of varying difficulty. This benchmark suite is available via COCO platform<sup>1</sup>. We ran experiments with multiple problems from this benchmark, but for brevity, we present canonical results on four benchmarks (shown in Table 1) noting that all the results show similar trends.

**Real world benchmarks.** We employ six diverse real-world domains. The complete details (function definition, bounds for input variables etc.) are in the Appendix.

**1) Pressure vessel design optimization.** This mechanical design problem (Kannan & Kramer, 1994; Tanabe & Ishibuchi, 2020) involves minimizing the total cost of a cylindrical pressure vessel. There are two discrete (thickness of shell and head of pressure vessel) and two continuous (inner radius and length of cylindrical section) variables.

**2) Welded beam design optimization.** The goal in this material engineering domain (Deb & Goyal, 1996; Reklaitis et al., 1983) is to design a welded beam while minimizing the overall cost of the fabrication. There are six variables: two discrete (type of welding configuration and bulk material of the beam) and four continuous (weld thickness, welded joint length, beam width and thickness).

<sup>1</sup><https://github.com/numbbo/coco>

**3) Speed reducer design optimization.** In this domain from NASA (Cagnina et al., 2008), the goal is to minimize the weight of a speed reducer defined over seven input variables: one discrete (number of teeth on pinion) and six continuous (face width, teeth module, lengths of shafts between bearings, and diameters of the shafts)

**4) Optimizing control for robot pushing.** This is a 14 dimensional control parameter tuning problem, where a robot is trying to push objects toward a goal location (Wang et al., 2018). We consider a hybrid version of this problem by discretizing ten input variables corresponding to location of the robot and number of simulation steps. The remaining four parameters corresponding to rotation are kept as continuous.

**5) Calibration of environmental model.** The problem of calibration and uncertainty analysis of expensive environmental models is very important in scientific domains (Bliznyuk et al., 2008; Astudillo & Frazier, 2019). There are four input variables (one discrete and three continuous).

**6) Hyper-parameter optimization.** We consider hyper-parameter tuning of a neural network model on a diverse set of benchmarks (Gijsbers et al., 2019): five discrete (hidden layer size, activation type, batch size, type of learning rate, and whether to use early stopping or not) and three continuous (learning rate initialization, momentum parameter, and regularization coefficient) hyper-parameters.

## 5.2. Experimental Setup

**Baseline methods.** We compare HyBO with four strong baselines: 1) CoCaBO, a state-of-the-art method (Ru et al., 2020); 2) SMAC (Hutter et al., 2010); 3) TPE (Bergstra et al., 2011); 4) HyBO w/o Marg is a special case of HyBO, where we do not perform marginalization over the hyper-parameters of the hybrid diffusion kernel; and 5) Cont-BO treats discrete variables as continuous and performs standard BO over continuous spaces (both modeling and acquisition function optimization). We did not include MiVaBO (Daxberger et al., 2020) as there was no publicly available implementation (Daxberger)<sup>2</sup>.

**Configuration of algorithms and baselines.** We configure HyBO as follows. We employ uniform prior for the length scale hyperparameter ( $\sigma$ ) of the RBF kernel. Horse-shoe prior is used for  $\beta$  hyper-parameter of the discrete diffusion kernel (Equation 4.8) and hyper-parameters  $\theta$  of the additive diffusion kernel (Equation 4.9). We employ expected improvement (Mockus et al., 1978) as the acquisition function. For acquisition function optimization, we perform iterative search over continuous and discrete sub-spaces as shown in Algorithm 1. For optimizing discrete subspace, we run local search with 20 restarts. We normalize each continuous variable to be in the range  $[-1, 1]$  and employed CMA-ES

algorithm<sup>3</sup> for optimizing the continuous subspace. We found that the results obtained by CMA-ES were not sensitive to its hyper-parameters. Specifically, we fixed the population size to 50 and initial standard deviation to 0.1 in all our experiments. We employed the open-source python implementation of CoCaBO<sup>4</sup>, SMAC<sup>5</sup>, and TPE<sup>6</sup>.

All the methods are initialized with same random hybrid structures. We replicated all experiments for 25 different random seeds and report the mean and two times the standard error in all our figures.

**Evaluation metric.** We use the best function value achieved after a given number of iterations (function evaluations) as a metric to evaluate all methods. The method that uncovers high-performing hybrid structures with less number of function evaluations is considered better.

## 5.3. Results and Discussion

**Results on mixed integer benchmark suite.** Figure 1 shows the canonical results on four benchmarks from bbox-mixint listed in Table 1 noting that all results show similar trends. HyBO and its variant HyBO-Round performs significantly better and converges much faster than all the other baselines. One key reason for this behavior is that hybrid diffusion kernel accounts for higher-order interactions between variables. Cont-BO performs the worst among all the methods. This shows that simply treating discrete variables as continuous is sub-optimal and emphasizes the importance of modeling the structure in discrete variables.

**Ablation results for statistical models.** To understand the reasons for the better performance of HyBO, we compare the performance of its surrogate model based on hybrid diffusion kernels with those of CoCaBO and SMAC. We perform the following experiment. We constructed testing dataset (pairs of hybrid structures and their function evaluations) of size 200 via uniform random sampling. We compute the mean absolute error (MAE) of the three surrogate models as a function of training set size. The results are shown in Figure 2 which depicts the mean and two times standard error of the MAE on 25 random testing datasets. HyBO clearly has very low error compared to CoCaBO and SMAC on Function 1 and 2. Although HyBO has similar MAE to CoCaBO in the beginning on Function 3 and 4, it rapidly decreases as the training set size increases which is not the case for other two methods. This experiment provides strong empirical evidence for the fact that the proposed surrogate model in HyBO can model hybrid spaces more accurately when compared to CoCaBO and SMAC.

**Ablation results for marginalization in HyBO.** Bayesian

<sup>3</sup><https://github.com/CMA-ES/pycma>

<sup>4</sup>[https://github.com/rubinxin/CoCaBO\\_code](https://github.com/rubinxin/CoCaBO_code)

<sup>5</sup><https://github.com/automl/SMAC3>

<sup>6</sup><https://github.com/hyperopt/hyperopt>

<sup>2</sup>Personal communication with the lead author.

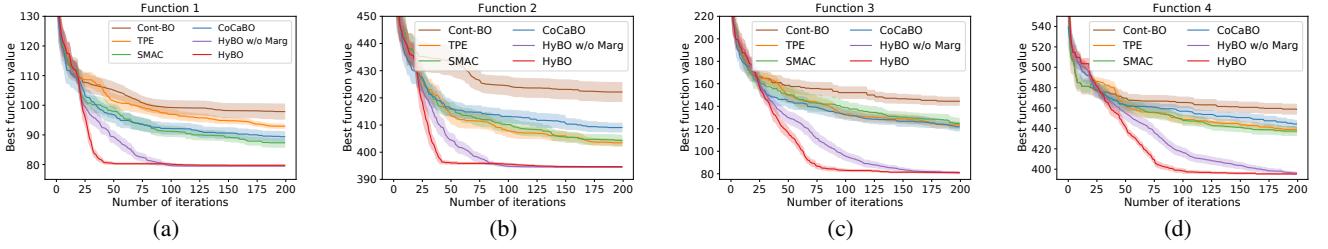


Figure 1. Results of HyBO and state-of-the-art baselines on bbo-mixint benchmark suite for functions shown in Table 1.

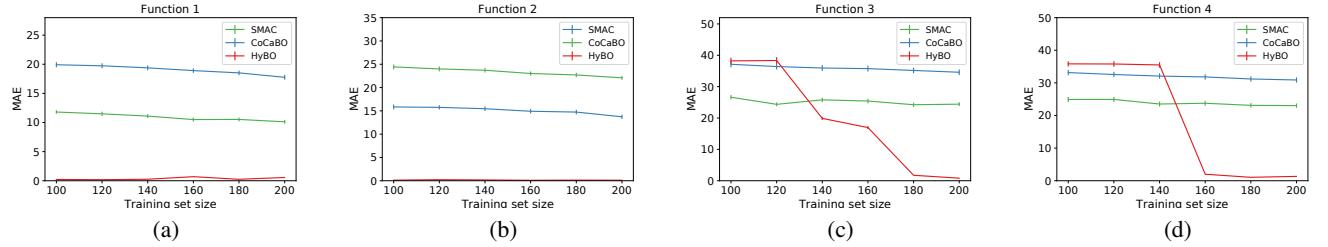


Figure 2. Results showing mean absolute test error with increasing size of training set on the bbo-mixint synthetic benchmarks.

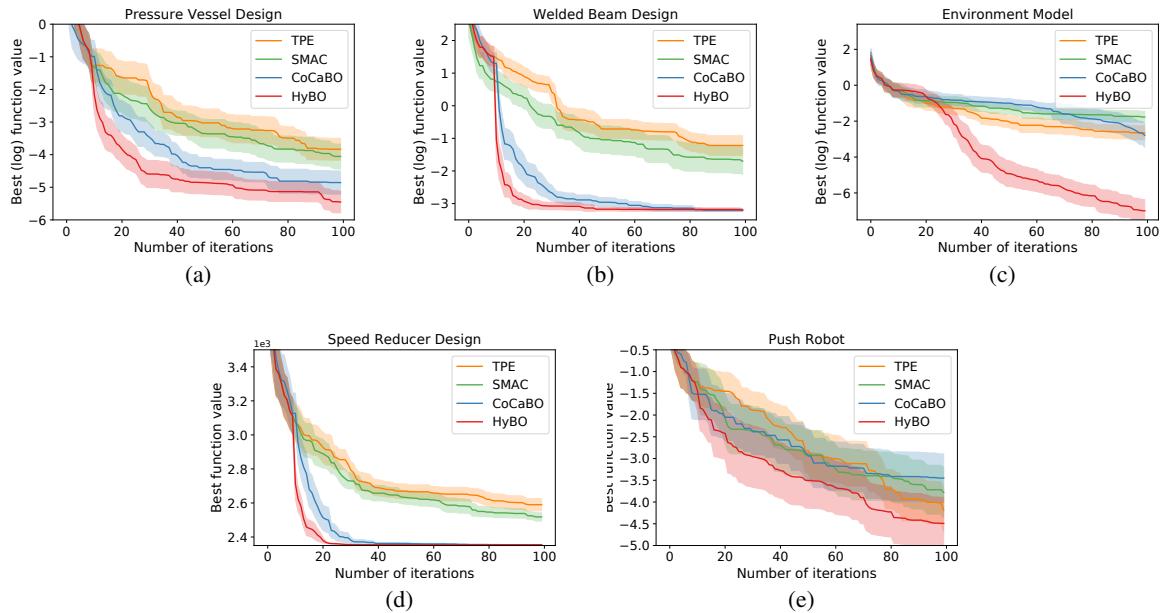


Figure 3. Results comparing the proposed HyBO approach with state-of-the-art baselines on multiple real world benchmarks.

treatment of hyper-parameters (marginalization) is one key component of our proposed HyBO method. However, to decouple the efficacy of additive diffusion kernel from the usage of marginalization, we performed experiments using HyBO without marginalization (HyBO w/o Marg in Figures). As evident from Figure 1, HyBO w/o Marg finds better solutions than all the baselines albeit with slower convergence which is improved by adding marginalization.

**Results for real-world domains.** Figure 3 shows comparison of HyBO approach with baseline methods on all real-world domains except hyper-parameter optimization. We make the following observations. 1) HyBO consistently performs better than all the baselines on all these benchmarks. 2) Even on benchmarks such as speed reducer design and welded beam design where HyBO finds a similar solution as CoCaBO, it does so with much faster convergence. 3)

| Dataset           | Cont-BO        | TPE            | SMAC           | CoCaBO         | HyBO                  |
|-------------------|----------------|----------------|----------------|----------------|-----------------------|
| blood_transfusion | 76.089 (0.325) | 76.711 (0.432) | 76.658 (0.418) | 76.978 (0.455) | <b>77.819 (0.463)</b> |
| kc1               | 85.185 (0.129) | 85.637 (0.069) | 85.453 (0.087) | 85.415 (0.099) | 85.466 (0.116)        |
| vehicle           | 80.501 (1.120) | 80.913 (1.051) | 83.669 (1.013) | 82.882 (1.222) | <b>86.104 (0.894)</b> |
| segment           | 87.253 (0.995) | 87.792 (0.537) | 89.986 (0.692) | 89.639 (0.727) | <b>91.433 (0.277)</b> |
| cnae              | 95.370 (0.103) | 95.691 (0.082) | 95.605 (0.063) | 95.679 (0.108) | 95.644 (0.135)        |
| jasmine           | 77.317 (0.216) | 77.893 (0.071) | 77.460 (0.189) | 77.513 (0.202) | 77.121 (0.172)        |

Table 2. Results on the task of hyper-parameter tuning of neural network models. Bold numbers signify statistical significance.

| Benchmark              | TPE   | SMAC | CoCaBO | HyBO |
|------------------------|-------|------|--------|------|
| Synthetic Function 1   | 0.012 | 2.34 | 2.30   | 50   |
| Synthetic Function 2   | 0.012 | 0.98 | 1.31   | 50   |
| Synthetic Function 3   | 0.026 | 2.99 | 3.18   | 180  |
| Synthetic Function 4   | 0.026 | 1.98 | 2.96   | 180  |
| Pressure Vessel Design | 0.003 | 0.34 | 0.85   | 20   |
| Welded Beam Design     | 0.004 | 0.64 | 1.02   | 40   |
| Speed Reducer Design   | 0.006 | 1.38 | 0.94   | 40   |
| Push Robot             | 0.017 | 1.94 | 1.70   | 90   |
| Environment model      | 0.005 | 0.31 | 0.50   | 40   |

Table 3. Computational cost in average wall-clock time (seconds) per BO iteration.

CoCaBO performs reasonably well on these benchmarks but its performance is worse than HyBO demonstrating that its sum kernel (along with Hamming kernel for discrete spaces) is less powerful than hybrid diffusion kernel of HyBO. 4). TPE has the worst performance on most benchmarks possibly a direct result of its drawback of not modeling the interactions between input dimensions. 5) SMAC performs poorly on all the benchmarks potentially due to poor uncertainty estimates from random forest surrogate model.

Table 2 shows the final accuracy (mean and standard error) obtained by all methods including HyBO on the task of tuning neural network models for six different datasets (BO curves are similar for all methods). HyBO produces comparable or better results than baseline methods.

**Computational cost analysis.** We compare the runtime of different algorithms including HyBO. All experiments were run on a AMD EPYC 7451 24-Core machine. Table 3 shows the average wall-clock time (in seconds) per BO iteration. We can see that HyBO is relatively expensive when compared to baseline methods. However, for real-world science and engineering applications, minimizing the cost of physical resources to perform evaluation (e.g., conducting an additive manufacturing experiment for designing materials such as alloys) is the most important metric. The computational cost for selecting inputs for evaluation is a secondary concern. HyBO uses more time to select inputs for evaluation to minimize the number of function evaluations to uncover better structures. We provide a finer-analysis of the HyBO runtime in Table 4. Each kernel evaluation time with all orders of interactions is very small. The overall runtime

is spent on two major things: a) Sampling from posterior distributions of hyperparameters using slice sampling; and b) AFO using CMA-ES + local search. We can reduce the sampling time by considering HyBO without marginalization which shows slightly worse performance, but takes only 10 percent of the sampling time in HyBO.

| Orders of interaction | HyBO iteration | AFO | Sampling | Kernel eval. |
|-----------------------|----------------|-----|----------|--------------|
| 2                     | 62             | 46  | 16       | 0.005        |
| 5                     | 68             | 50  | 18       | 0.006        |
| 10                    | 102            | 68  | 34       | 0.010        |
| 20 (HyBO)             | 180            | 114 | 66       | 0.020        |

Table 4. Average runtime (seconds) for different orders of interaction within hybrid kernel for synthetic Function 3.

## 6. Conclusions

We studied a novel Bayesian optimization approach referred as HyBO for optimizing hybrid spaces using Gaussian process based surrogate models. We presented a principled approach to construct hybrid diffusion kernels by combining diffusion kernels defined over continuous and discrete sub-spaces in a tractable and flexible manner to capture the interactions between discrete and continuous variables. We proved that additive hybrid kernels have the universal approximation property. Our experimental results on diverse synthetic and real-world benchmarks show that HyBO performs significantly better than state-of-the-art methods.

**Acknowledgements.** This research is supported by NSF grants IIS-1845922, OAC-1910213, and CNS-1955353.

## References

- Astudillo, R. and Frazier, P. Bayesian optimization of composite functions. volume 97 of *Proceedings of Machine Learning Research*, pp. 354–363. PMLR, 09–15 Jun 2019.
- Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing*, 32(1):48–77, 2002.
- Baptista, R. and Poloczek, M. Bayesian optimization of combinatorial structures. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 462–471, 2018.
- Belakaria, S., Deshwal, A., and Doppa, J. R. Max-value entropy search for multi-objective Bayesian optimization. In *NeurIPS*, 2019.
- Belakaria, S., Deshwal, A., and Doppa, J. R. Max-value entropy search for multi-objective Bayesian optimization with constraints. *CoRR*, abs/2009.01721, 2020a. URL <https://arxiv.org/abs/2009.01721>.
- Belakaria, S., Deshwal, A., and Doppa, J. R. Multi-fidelity multi-objective Bayesian optimization: An output space entropy search approach. In *AAAI conference on Artificial Intelligence (AAAI)*, 2020b.
- Belakaria, S., Deshwal, A., and Doppa, J. R. Uncertainty aware search framework for multi-objective Bayesian optimization with constraints. *CoRR*, abs/2008.07029, 2020c. URL <https://arxiv.org/abs/2008.07029>.
- Belakaria, S., Deshwal, A., and Doppa, J. R. Information-theoretic multi-objective Bayesian optimization with continuous approximations. *CoRR*, abs/2009.05700, 2020d. URL <https://arxiv.org/abs/2009.05700>.
- Belakaria, S., Deshwal, A., Jayakodi, N. K., and Doppa, J. R. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *AAAI*, 2020e.
- Belakaria, S., Jackson, D., Cao, Y., Doppa, J. R., and Lu, X. Machine learning enabled fast multi-objective optimization for electrified aviation power system design. In *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2020f.
- Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 2546–2554, 2011.
- Bliznyuk, N., Ruppert, D., Shoemaker, C., Regis, R., Wild, S., and Mugunthan, P. Bayesian calibration and uncertainty analysis for computationally expensive models using optimization and radial basis function approximation. *Journal of Computational and Graphical Statistics*, 17(2):270–294, 2008.
- Cagnina, L., Esquivel, S., and Coello, C. Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica*, 32:319–326, 2008.
- Chung, F. R. and Graham, F. C. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.
- Daxberger, E. Personal communication about MiVaBO implementation and code.
- Daxberger, E., Makarova, A., Turchetta, M., and Krause, A. Mixed-variable bayesian optimization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 2633–2639, 7 2020.
- Deb, K. and Goyal, M. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.
- Deshwal, A., Belakaria, S., and Doppa, J. R. Scalable combinatorial Bayesian optimization with tractable statistical models. *CoRR*, abs/2008.08177, 2020a. URL <https://arxiv.org/abs/2008.08177>.
- Deshwal, A., Belakaria, S., Doppa, J. R., and Fern, A. Optimizing discrete spaces via expensive evaluations: A learning to search framework. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020b.
- Deshwal, A., Belakaria, S., and Doppa, J. R. Mercer features for efficient combinatorial Bayesian optimization. In *AAAI*, 2021.
- Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. Additive gaussian processes. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 226–234, 2011.
- Falkner, S., Klein, A., and Hutter, F. BOHB: robust and efficient hyper-parameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 1436–1445, 2018.
- Frazier, P. I. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.
- Garrido-Merchán, E. C. and Hernández-Lobato, D. Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35, 2020.

- Gijsbers, P., LeDell, E., Poirier, S., Thomas, J., Bischl, B., and Vanschoren, J. An open source automl benchmark. *arXiv preprint arXiv:1907.00909.*, 2019. Accepted at AutoML Workshop at ICML 2019.
- Greenhill, S., Rana, S., Gupta, S., Vellanki, P., and Venkatesh, S. Bayesian optimization for adaptive experimental design: A review. *IEEE Access*, 8:13937–13948, 2020.
- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.
- Hansen, N. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.
- Hutter, F., Hoos, H. H., and Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration (extended version). Technical Report TR-2010-10, University of British Columbia, Department of Computer Science, 2010. Available online: <http://www.cs.ubc.ca/~hutter/papers/10-TR-SMAC.pdf>.
- Hutter, F., Hoos, H. H., and Leyton-Brown, K. Sequential model-based optimization for general algorithm configuration. In *International conference on Learning and Intelligent Optimization*, pp. 507–523, 2011.
- Kannan, B. K. and Kramer, S. N. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, 116(2):405–411, 06 1994.
- Kondor, R. and Lafferty, J. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2002, pp. 315–322, 2002.
- Kondor, R. and Vert, J.-P. Diffusion kernels. *Kernel methods in computational biology*, pp. 171–192, 2004.
- Lafferty, J. and Lebanon, G. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6(1): 129–163, 2005.
- Li, L., Jamieson, K. G., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. Hyperband: a novel bandit-based approach to hyper-parameter optimization. *Journal of Machine Learning Research (JMLR)*, 18:185:1–185:52, 2017.
- Mania, H., Ramdas, A., Wainwright, M. J., Jordan, M. I., and Recht, B. On kernel methods for covariates that are rankings. *Electronic Journal of Statistics*, 12(2):2537–2577, 2018.
- Micchelli, C. A., Xu, Y., and Zhang, H. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.
- Mockus, J., Tiesis, V., and Zilinskas, A. The application of bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129), 1978.
- Neal, R. M. Slice sampling. *Annals of statistics*, 31(3): 705–741, 6 2003.
- Oh, C., Tomczak, J., Gavves, E., and Welling, M. Combinatorial bayesian optimization using the graph cartesian product. In *Advances in Neural Information Processing Systems*, pp. 2910–2920, 2019.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20*, pp. 1177–1184, 2007.
- Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.
- Reklaitis, G. V., Ravindran, A., and Ragsdell, K. M. *Engineering optimization: methods and applications*. Wiley New York, 1983.
- Roustant, O., Padonou, E., Deville, Y., Clément, A., Perrin, G., Giorla, J., and Wynn, H. Group kernels for gaussian process metamodels with categorical inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2):775–806, 2020.
- Ru, B., Alvi, A. S., Nguyen, V., Osborne, M. A., and Roberts, S. J. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning (ICML)*, 2020.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.
- Sriperumbudur, B. K., Fukumizu, K., and Lanckriet, G. R. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(7), 2011.
- Steinwart, I. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2(Nov):67–93, 2001.

Steinwart, I. and Christmann, A. *Support vector machines*. Springer Science & Business Media, 2008.

Steinwart, I., Thomann, P., and Schmid, N. Learning with hierarchical gaussian kernels. *arXiv preprint arXiv:1612.00824*, 2016.

Tanabe, R. and Ishibuchi, H. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020.

Tušar, T., Brockhoff, D., and Hansen, N. Mixed-integer benchmark problems for single-and bi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 718–726, 2019.

Verel, S., Derbel, B., Liefoghe, A., Aguirre, H., and Tanaka, K. A surrogate model based on walsh decomposition for pseudo-boolean functions. In *International Conference on Parallel Problem Solving from Nature*, pp. 181–193. Springer, 2018.

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

Zhou, Z., Belakaria, S., Deshwal, A., Hong, W., Doppa, J. R., Pande, P. P., and Heo, D. Design of multi-output switched-capacitor voltage regulator via machine learning. In *DATE*, 2020.