Mixed Variable Bayesian Optimization with Frequency Modulated Kernels

Changyong Oh[1] **Efstratios Gavves**[1] **Max Welling**[1,2]

[1]Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
[2]Qualcomm AI Research Netherlands, Amsterdam, The Netherlands

Preliminary version

Abstract

The sample efficiency of Bayesian optimization (BO) is often boosted by Gaussian Process (GP) surrogate models. However, on mixed variable spaces, surrogate models other than GPs are prevalent, mainly due to the lack of kernels which can model complex dependencies across different types of variables. In this paper, we propose the frequency modulated (FM) kernel flexibly modeling dependencies among different types of variables, so that BO can enjoy the further improved sample efficiency. The FM kernel uses distances on continuous variables to modulate the graph Fourier spectrum derived from discrete variables. However, the frequency modulation does not always define a kernel with the similarity measure behavior which returns higher values for pairs of more similar points. Therefore, we specify and prove conditions for FM kernels to be positive definite and to exhibit the similarity measure behavior. In experiments, we demonstrate the improved sample efficiency of GP BO using FM kernels (BO-FM). On synthetic problems and hyperparameter optimization problems, BO-FM outperforms competitors consistently. Also, the importance of the frequency modulation principle is empirically demonstrated on the same problems. On joint optimization of neural architectures and SGD hyperparameters, BO-FM outperforms competitors including Regularized evolution (RE) and BOHB. Remarkably, BO-FM performs better even than RE and BOHB using three times as many evaluations.

1 INTRODUCTION

Bayesian optimization has found many applications ranging from daily routine level tasks of finding a tasty cookie recipe [Solnik et al., 2017] to sophisticated hyperparameter optimization tasks of machine learning algorithms (e.g. Alpha-Go [Chen et al., 2018]). Much of this success is attributed to the flexibility and the quality of uncertainty quantification of Gaussian Process (GP)-based surrogate models [Snoek et al., 2012, Swersky et al., 2013, Oh et al., 2018].

Despite the superiority of GP surrogate models, as compared to non-GP ones, their use on spaces with discrete structures (*e.g.,* chemical spaces [Reymond and Awale, 2012], graphs and even mixtures of different types of spaces) is still application-specific [Kandasamy et al., 2018, Korovina et al., 2019]. The main reason is the difficulty of defining kernels flexible enough to model dependencies across different types of variables. On mixed variable spaces which consist of different types of variables including continuous, ordinal and nominal variables, current BO approaches resort to non-GP surrogate models, such as simple linear models or linear models with manually chosen basis functions [Daxberger et al., 2019]. However, such linear approaches are limited because they may lack the necessary model capacity.

There is much progress on BO using GP surrogate models (GP BO) for continuous, as well as for discrete variables. However, for mixed variables it is not straightforward how to define kernels ,which can model dependencies across different types of variables. To bridge the gap, we propose *frequency modulation* which uses distances on continuous variables to modulate the frequencies of the graph spectrum [Ortega et al., 2018] where the graph represents the discrete part of the search space [Oh et al., 2019].

A potential problem in the frequency modulation is that it does not always define a kernel with the similarity measure behavior [Vert et al., 2004]. That is, the frequency modulation does not necessarily define a kernel that returns higher values for pairs of more similar points. Formally, for a stationary kernel $k(x,y) = s(x-y)$, s should be decreasing [Remes et al., 2017]. In order to guarantee the similarity measure behavior of kernels constructed by frequency modulation, we stipulate a condition, the *frequency modula-*

Accepted for the 37^th^ Conference on Uncertainty in Artificial Intelligence (UAI 2021).

tion principle. Theoretical analysis results in proofs of the positive definiteness as well as the effect of the frequency modulation principle. We coin frequency modulated (FM) kernels as the kernels constructed by frequency modulation and respecting the frequency modulation principle.

Different to methods that construct kernels on mixed variables by kernel addition and kernel multiplication, for example, FM kernels do not impose an independence assumption among different types of variables. In FM kernels, quantities in the two domains, that is the distances in a spatial domain and the frequencies in a Fourier domain, interact. Therefore, the restrictive independence assumption is circumvented, and thus flexible modeling of mixed variable functions is enabled.

In this paper, *(i)* we propose frequency modulation, a new way to construct kernels on mixed variables, *(ii)* we provide the condition to guarantee the similarity measure behavior of FM kernels together with a theoretical analysis, and *(iii)* we extend frequency modulation so that it can model complex dependencies between arbitrary types of variables. In experiments, we validate the benefit of the increased modeling capacity of FM kernels and the importance of the frequency modulation principle for improved sample efficiency on different mixed variable BO tasks. We also test BO with GP using FM kernels (BO-FM) on a challenging joint optimization of the neural architecture and the hyperparameters with two strong baselines, Regularized Evolution (RE) [Real et al., 2019] and BOHB [Falkner et al., 2018]. BO-FM outperforms both baselines which have proven their competence in neural architecture search [Dong et al., 2021]. Remarkably, BO-FM outperforms RE with three times evaluations.

2 PRELIMINARIES

2.1 BAYESIAN OPTIMIZATION WITH GAUSSIAN PROCESSES

Bayesian optimization (BO) aims at finding the global optimum of a black-box function g over a search space $\mathscr{X}$. At each round BO performs an evaluation y_i on a new point $\mathbf{x}_i \in \mathscr{X}$, collecting the set of evaluations $\mathscr{D}_t = \{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,t}$ at the t-th round. Then, a surrogate model approximates the function g given $\mathscr{D}_t$ using the predictive mean $\mu(\mathbf{x}_* \,|\, \mathscr{D}_t)$ and the predictive variance $\sigma^2(\mathbf{x}_* \,|\, \mathscr{D}_t)$. Now, an acquisition function $r(\mathbf{x}_*) = r(\mu(\mathbf{x}_* \,|\, \mathscr{D}_t), \sigma^2(\mathbf{x}_* \,|\, \mathscr{D}_t))$ quantifies how informative input $\mathbf{x} \in \mathscr{X}$ is for the purpose of finding the global optimum. g is then evaluated at $\mathbf{x}_{t+1} = \mathrm{argmax}_{\mathbf{x}\in\mathscr{X}}\, r(\mathbf{x})$, $y_{t+1} = g(\mathbf{x}_{t+1})$. With the updated set of evaluations, $\mathscr{D}_{t+1} = \mathscr{D}_t \cup \{(\mathbf{x}_{t+1}, y_{t+1})\}$, the process is repeated.

A crucial component in BO is thus the surrogate model. Specifically, the quality of the predictive distribution of the surrogate model is critical for balancing the exploration-exploitation trade-off [Shahriari et al., 2015]. Compared with other surrogate models (such as Random Forest [Hutter et al., 2011] and a tree-structured density estimator [Bergstra et al., 2011]), Gaussian Processes (GPs) tend to yield better results [Snoek et al., 2012, Oh et al., 2018].

For a given kernel k and data $\mathscr{D} = (\mathbf{X}, \mathbf{y})$ where $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^T$ and $\mathbf{y} = [y_1, \cdots, y_n]^T$, a GP has a predictive mean $\mu(\mathbf{x}_* | \mathbf{X}, \mathbf{y}) = k_{*\mathbf{X}}(k_{\mathbf{XX}} + \sigma^2 I)^{-1}\mathbf{y}$ and predictive variance $\sigma^2(\mathbf{x}_* | \mathbf{X}, \mathbf{y}) = k_{**} - k_{*\mathbf{X}}(k_{\mathbf{XX}} + \sigma^2 I)^{-1} k_{\mathbf{X}*}$ where $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$, $[k_{*\mathbf{X}}]_{1,i} = k(\mathbf{x}_*, \mathbf{x}_i)$, $k_{\mathbf{X}*} = (k_{*\mathbf{X}})^T$ and $[k_{\mathbf{XX}}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$.

2.2 KERNELS ON DISCRETE VARIABLES

We first review some kernel terminology [Scholkopf and Smola, 2001] that is needed in the rest of the paper.

Definition 2.1 (Gram Matrix)**.** Given a function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ and data $x_1, \cdots, x_n \in \mathscr{X}$, the $n \times n$ matrix K with elements $[K]_{ij} = k(x_i, x_j)$ is called the Gram matrix of k with respect to $x_1, \cdots, x_n$.

Definition 2.2 (Positive Definite Matrix)**.** A real $n \times n$ matrix K satisfying $\sum_{i,j} a_i [K]_{ij} a_j \geq 0$ for all $a_i \in \mathbb{R}$ is called positive definite (PD)[1].

Definition 2.3 (Positive Definite Kernel)**.** A function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ which gives rise to a positive definite Gram matrix for all $n \in \mathbb{N}$ and all $x_1, \cdots, x_n \in \mathscr{X}$ is called a positive definite (PD) kernel, or simply a kernel.

A search space which consists of discrete variables, including both nominal and ordinal variables, can be represented as a graph [Kondor and Lafferty, 2002, Oh et al., 2019]. In this graph each vertex represents one state of exponentially many joint states of the discrete variables. The edges represent relations between these states (e.g. if they are similar) [Oh et al., 2019]. With a graph representing a search space of discrete variables, kernels on a graph can be used for BO. In [Smola and Kondor, 2003], for a positive decreasing function f and a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ whose graph Laplacian $L(\mathscr{G})$[2] has the eigendecomposition $U \Lambda U^T$, it is shown that a kernel can be defined as

$$k_{disc}(v, v' | \beta) = [U f(\Lambda | \beta) U^T]_{v,v'} \tag{1}$$

where $\beta \geq 0$ is a kernel parameter and f is a positive decreasing function. It is the reciprocal of a regularization operator [Smola and Kondor, 2003] which penalizes high frequency components in the spectrum.

[1]Sometimes, different terms are used, semi-positive definite for $\sum_{i,j} a_i [K]_{ij} a_j \geq 0$ and positive definite for $\sum_{i,j} a_i [K]_{ij} a_j > 0$. Here, we stick to the definition in [Scholkopf and Smola, 2001].

[2]In this paper, we use a (unnormalized) graph Laplacian $L(\mathscr{G}) = D - A$ while, in [Smola and Kondor, 2003], symmetric normalized graph Laplacian, $L^{sym}(\mathscr{G}) = D^{-1/2}(D-A)D^{-1/2}$. ($A$: adj. mat. / D : deg. mat.) Kernels are defined for both.

Preliminary version

3 MIXED VARIABLE BAYESIAN OPTIMIZATION

With the goal of obtaining flexible kernels on mixed variables which can model complex dependencies across different types of variables, we propose the frequency modulated (FM) kernel. Our objective is to enhance the modelling capacity of GP surrogate models and, thereby improve the sample efficiency of mixed-variable BO. FM kernels use the continuous variables to modulate the frequencies of the kernel of discrete variables defined on the graph. As a consequence, FM kernels can model complex dependencies between continuous and discrete variables. Specifically, let us start with continuous variables of dimension $D_{\mathscr{C}}$, and discrete variables represented by the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ whose graph Laplacian $L(\mathscr{G})$ has eigendecompostion $U\Lambda U^T$. To define a frequency modulated kernel we consider the function $k : (\mathbb{R}^{D_{\mathscr{C}}} \times \mathscr{V}) \times (\mathbb{R}^{D_{\mathscr{C}}} \times \mathscr{V}) \Rightarrow \mathbb{R}$ of the following form

$$\begin{aligned} &k((\mathbf{c}, v), (\mathbf{c}', v') | \beta, \theta) \\ &= \sum_{i=1}^{|\mathscr{V}|} [U]_{v,i} f(\lambda_i, \| \mathbf{c} - \mathbf{c}' \|_\theta | \beta) [U]_{v',i} \end{aligned} \tag{2}$$

where $\| \mathbf{c} - \mathbf{c}' \|_\theta^2 = \sum_{d=1}^{D_{\mathscr{C}}} (c_d - c_d')^2 / \theta_d^2$ and (θ, β) are tunable parameters. f is the frequency modulating function defined below in Def. 3.1.

The function f in Eq. (2) takes frequency λ_i and distance $\| \mathbf{c} - \mathbf{c}' \|_\theta^2$ as arguments, and its output is combined with the basis $[U]_{v,i}$. That is, the function f processes the information in each eigencomponent separately while Eq. (2) then sums up the information processed by f. Note that unlike kernel addition and kernel product,[3] the distance $\| \mathbf{c} - \mathbf{c}' \|_\theta^2$ influences each eigencomponent separately as illustrated in Figure.1. Unfortunately, Eq. (2) with an arbitrary function f does not always define a positive definite kernel. Moreover, Eq. (2) with an arbitrary function f may return higher kernel values for less similar points, which is not expected from a proper similarity measure [Vert et al., 2004]. To this end, we first specify three properties of functions f such that Eq. (2) guaranteed to be a positive definite kernel and a proper similarity measure at the same time. Then, we motivate the necessity of each of the properties in the following subsections.

Definition 3.1 (Frequency modulating function)**.** A frequency modulating function is a function $f : \mathbb{R}^+ \times \mathbb{R} \rightarrow \mathbb{R}$ satisfying the three properties below.

FM-P1 For a fixed $t \in \mathbb{R}$, $f(s,t)$ is a positive and decreasing function with respect to s on $[0, \infty)$.

FM-P2 For a fixed $s \in \mathbb{R}^+$, $f(s, \| \mathbf{c} - \mathbf{c}' \|_\theta)$ is a positive definite kernel on $(\mathbf{c}, \mathbf{c}') \in \mathbb{R}^{D_{\mathscr{C}}} \times \mathbb{R}^{D_{\mathscr{C}}}$.

FM-P3 For $t_1 < t_2$, $h_{t_1,t_2}(s) = f(s,t_1) - f(s,t_2)$ is positive, strictly decreasing and convex w.r.t $s \in \mathbb{R}^+$.

[3] e.g $k_{add}((\mathbf{c}, v), (\mathbf{c}', v')) = e^{-\| \mathbf{c} - \mathbf{c}' \|_\theta^2} + k_{disc}(v, v')$ and $k_{prod}((\mathbf{c}, v), (\mathbf{c}', v')) = e^{-\| \mathbf{c} - \mathbf{c}' \|_\theta^2} \cdot k_{disc}(v, v')$

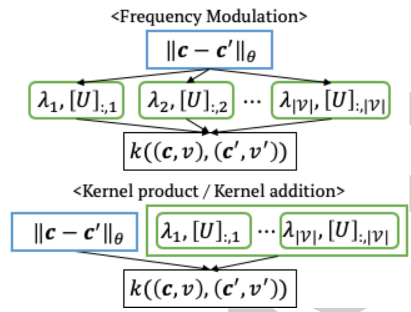


Figure 1: Influence on eigencomponents

Definition 3.2 (FM kernel)**.** A FM kernel is a function on $(\mathbb{R}^{D_{\mathscr{C}}} \times \mathscr{V}) \times (\mathbb{R}^{D_{\mathscr{C}}} \times \mathscr{V})$ of the form in Eq. (2), where f is a frequency modulating function on $\mathbb{R}^+ \times \mathbb{R}$.

3.1 FREQUENCY REGULARIZATION OF FM KERNELS

In [Smola and Kondor, 2003], it is shown that Eq. (1) defines a kernel that regularizes the eigenfunctions with high frequencies when f is positive and decreasing. It is also shown that the reciprocal of f in Eq. (1) is a corresponding regularization operator. For example, the diffusion kernel defined with $f(\lambda) = \exp(-\beta\lambda)$ corresponds to the regularization operator $r(\lambda) = \exp(\beta\lambda)$. The regularized Laplacian kernel defined with $f(\lambda) = 1/(1 + \beta\lambda)$ corresponds to the regularization operator $r(\lambda) = 1 + \beta\lambda$. Both regularization operators put more penalty on higher frequencies λ.

Therefore, the property **FM-P1** forces FM kernels to have the same regularization effect of promoting a smoother function by penalizing the eigenfunctions with high frequencies.

3.2 POSITIVE DEFINITENESS OF FM KERNELS

Determining whether Eq.2 defines a positive definite kernel is not trivial. The reason is that the gram matrix $[k((\mathbf{c}_i, v_i), (\mathbf{c}_j, v_j))]_{i,j}$ is not determined only by the entries v_i and v_j, but these entries are additionally affected by different distance terms $\| \mathbf{c}_i - \mathbf{c}_j \|_\theta$. To show that FM kernels are positive definite, it is sufficient to show that $f(\lambda_i, \| \mathbf{c} - \mathbf{c}' \|_\theta | \beta)$ is positive definite on $(\mathbf{c}, \mathbf{c}') \in \mathbb{R}^{D_{\mathscr{C}}} \times \mathbb{R}^{D_{\mathscr{C}}}$.

Theorem 3.1. *If $f(\lambda, \| \mathbf{c} - \mathbf{c}' \|_\theta | \beta)$ defines a positive definite kernel with respect to $\mathbf{c}$ and $\mathbf{c}'$, then the FM kernel with such f is positive definite **jointly** on $\mathbf{c}$ and v. That is, the positive definiteness of $f(\lambda, \| \mathbf{c} - \mathbf{c}' \|_\theta | \beta)$ on $\mathbb{R}^{D_{\mathscr{C}}}$ implies the positive definiteness of the FM kernel on $\mathbb{R}^{D_{\mathscr{C}}} \times \mathscr{V}$.*

Proof. See Supp. Sec.1, Thm. 1.1. □

Note that Theorem 3.1 shows that the property **FM-P2** guarantees that FMs kernels are positive definite jointly on $\mathbf{c}$ and v.

Preliminary version

In the current form of Theorem 3.1, the frequency modulating functions depend on the distance $\|\mathbf{c}-\mathbf{c}'\|_\theta$. However, the proof does not change for the more general form of $f(\lambda,\mathbf{c},\mathbf{c}'|\alpha,\beta)$, where f does not depend on $\|\mathbf{c}-\mathbf{c}'\|_\theta$. Hence, Theorem 3.1 can be extended to the more general case that $f(\lambda,\mathbf{c},\mathbf{c}'|\alpha,\beta)$ is positive definite on $(\mathbf{c},\mathbf{c}') \in \mathbb{R}^{D_\mathscr{C}} \times \mathbb{R}^{D_\mathscr{C}}$.

3.3 FREQUENCY MODULATION PRINCIPLE

A kernel, as a similarity measure, is expected to return higher values for pairs of more similar points and vice versa [Vert et al., 2004]. We call such behavior the *similarity measure behavior*.

In Eq. (2), the distance $\|\mathbf{c}-\mathbf{c}'\|_\theta$ represents a quantity in the "spatial" domain interacting with quantities λ_is in the "frequency" domain. Due to the interplay between the two different domains, the kernels of the form Eq. (2) do not exhibit the similarity measure behavior *for an arbitrary function f*. Next, we derive a sufficient condition on f for the similarity measure behavior to hold for FM kernels.

Formally, the similarity measure behavior is stated as

$$\begin{aligned}&\|\mathbf{c}-\mathbf{c}'\|_\theta \leq \|\tilde{\mathbf{c}}-\tilde{\mathbf{c}}'\|_\theta \\ &\Rightarrow\ k((\mathbf{c},v),(\mathbf{c}',v')) \geq k((\tilde{\mathbf{c}},v),(\tilde{\mathbf{c}}',v'))\end{aligned} \tag{3}$$

or equivalently,

$$\begin{aligned}&\|\mathbf{c}-\mathbf{c}'\|_\theta \leq \|\tilde{\mathbf{c}}-\tilde{\mathbf{c}}'\|_\theta \\ &\Rightarrow\ \sum_{i=1}^{|\mathscr{V}|} [U]_{v,i} h_{t_1,t_2}(\lambda_i|\beta) [U]_{v',i} \geq 0\end{aligned} \tag{4}$$

where $h_{t_1,t_2}(\lambda|\beta) = f(\lambda,t_1|\beta) - f(\lambda,t_2|\beta)$, $t_1 = \|\mathbf{c}-\mathbf{c}'\|_\theta$ and $t_2 = \|\tilde{\mathbf{c}}-\tilde{\mathbf{c}}'\|_\theta$.

Theorem 3.2. *For a connected and weighted undirected graph $\mathscr{G} = (\mathscr{V},\mathscr{E})$ with non-negative weights on edges, define a similarity (or kernel) $a(v,v') = [Uh(\Lambda)U^T]_{v,v'}$, where U and Λ are eigenvectors and eigenvalues of the graph Laplacian $L(\mathscr{G}) = U\Lambda U^T$. If h is any non-negative and strictly decreasing convex function on $[0,\infty)$, then $a(v,v') \geq 0$ for all $v,v' \in \mathscr{V}$.*

Therefore, these conditions on $h(\Lambda)$ result in a similarity measure a with only positive entries, which in turn proves property Eq. (4). Here, we provide a proof of the theorem for a simpler case with an unweighted complete graph, where Eq. (4) holds without the convexity condition on h.

Proof. For a unweighted complete graph with n vertices, we have eigenvalues $\lambda_1 = 0$, $\lambda_2 = \cdots = \lambda_n = n$ and eigenvectors such that $[U]_{\cdot 1} = 1/\sqrt{n}$ and $\sum_{i=1}^n [U]_{v,i}[U]_{v',i} = \delta_{vv'}$. For $v \neq v'$, the conclusion in Eq. (4), $\sum_{i=1}^n h(\lambda_i)[U]_{v,i}[U]_{v',i}$ becomes $h(0)/n + h(n)\sum_{i=2}^n [U]_{v,i}[U]_{v',i} = (h(0)-h(n))/n$ in which non-negativity follows with decreasing h.

For the complete proof, see Thm. 2.1 in Supp. Sec. 2. □

Theorem 3.2 thus shows that the property **FM-P3** is sufficient for Eq. (4) to hold. We call the property **FM-P3** the *frequency modulation principle*. Theorem 3.2 also implies the non-negativity of many kernels derived from graph Laplacian.

Corollary 3.2.1. *The random walk kernel derived from the symmetric normalized Laplacian [Smola and Kondor, 2003], the diffusion kernels [Kondor and Lafferty, 2002, Oh et al., 2019] and the regularized Laplacian kernel [Smola and Kondor, 2003] derived from symmetric normalized or unnormalized Laplacian, are all non-negatived valued.*

Proof. See Cor. 2.1.1 in Supp. Sec. 2. □

3.4 FM KERNELS IN PRACTICE

Scalability Since the (graph Fourier) frequencies and basis functions are computed by the eigendecomposition of cubic computational complexity, a plain application of frequency modulation makes the computation of FM kernels prohibitive for a large number of discrete variables. Given P discrete variables where each variable can be individually represented by a graph $\mathscr{G}_p$, the discrete part of the search space can be represented as a product space, $\mathscr{V} = \mathscr{V}_1 \times \cdots \times \mathscr{V}_P$.

In this case, we define FM kernels on $\mathbb{R}^{D_\mathscr{C}} \times \mathscr{V} = \mathbb{R}^{D_\mathscr{C}} \times (\mathscr{V}_1 \times \cdots \times \mathscr{V}_P)$ as

$$\begin{aligned}k((\mathbf{c},\mathbf{v}),(\mathbf{c}',\mathbf{v}')|\,\alpha,\beta,\theta) &= \prod_{p=1}^{P} k_p((\mathbf{c},v_p),(\mathbf{c}',v'_p)|\beta_p,\theta) \\ &= \prod_{p=1}^{P} \sum_{i=1}^{|\mathscr{V}_p|} [U^p]_{v_p,i} f(\lambda_i^p,\alpha_p\|\mathbf{c}-\mathbf{c}'\|_\theta|\beta_p)[U^p]_{v'_p,i}\end{aligned} \tag{5}$$

where $\mathbf{v} = (v_1,\cdots,v_P)$, $\mathbf{v}' = (v'_1,\cdots,v'_P)$, $\alpha = (\alpha_1,\cdots,\alpha_P)$ $\beta = (\beta_1,\cdots,\beta_P)$ and the graph Laplacian is given as $L(\mathscr{G}_p)$ with the eigendecomposition $U_p \mathrm{diag}[\lambda_1^p,\cdots,\lambda_{\|\mathscr{V}_p\|}^p]U_p^T$.

Eq.5 should not be confused with the kernel product of kernels on each $\mathscr{V}_p$. Note that the distance $\|\mathbf{c}-\mathbf{c}'\|_\theta$ is shared, which introduces the coupling among discrete variables and thus allows more modeling freedom than a product kernel. In addition to the coupling, the kernel parameter α_ps lets us individually determine the strength of the frequency modulation.

Examples Defining a FM kernel amounts to constructing a frequency modulating function. We introduce examples of flexible families of frequency modulating functions.

Preliminary version

Proposition 1. *For $S \in (0,\infty)$, a finite measure μ on $[0,S]$, μ-measurable $\tau : [0,S] \Rightarrow [0,2]$ and μ-measurable $\rho : [0,S] \Rightarrow \mathbb{N}$, the function of the form below is a frequency modulating function.*

$$
\begin{aligned}
& f(\lambda, \alpha \| \mathbf{c} - \mathbf{c}' \|_\theta | \beta) \\
&= \int_0^S \frac{1}{(1 + \beta\lambda + \alpha \| \mathbf{c} - \mathbf{c}' \|_\theta^{\tau(s)})^{\rho(s)}} \mu(ds)
\end{aligned} \tag{6}
$$

Proof. See Supp. Sec.3, Prop.1. □

Assuming $S = 1$ and $\tau(s) = 2$, Prop. 1 gives $(1 + \beta\lambda + \alpha \| \mathbf{c} - \mathbf{c}' \|_\theta^2)^{-1}$ with $\rho(s) = 1$ and $\mu(ds) = ds$, and $\sum_{n=1}^{N} a_n (1 + \beta\lambda + \alpha \| \mathbf{c} - \mathbf{c}' \|_\theta^2)^{-n}$ with $\rho(s) = \lfloor Ns \rfloor$ and $\mu(\{n/N\}) = a_n \geq 0$ and $\mu([\{n/N\}_{n=1,\cdots,N}^c) = 0$.

3.5 EXTENSION OF THE FREQUENCY MODULATION

Frequency modulation is not restricted to distances on Euclidean spaces but it is applicable to any arbitrary space with a kernel defined on it. As a concrete example of frequency modulation by kernels, we show a non-stationary extension where f does not depend on $\| \mathbf{c} - \mathbf{c}' \|_\theta$ but on the neural network kernel k_{NN} [Rasmussen, 2003]. Consider Eq. (2) with $f = f_{NN}$ as follows.

$$
f_{NN}(\lambda, k_{NN}(\mathbf{c}, \mathbf{c}' | \Sigma) | \beta) = \frac{1}{2 + \beta\lambda - k_{NN}(\mathbf{c}, \mathbf{c}' | \Sigma)} \tag{7}
$$

where $k_{NN}(\mathbf{c}, \mathbf{c}' | \Sigma) = \frac{2}{\pi} \arcsin\left(\frac{2 \mathbf{c}^T \Sigma \mathbf{c}'}{(1 + \mathbf{c}^T \Sigma \mathbf{c})(1 + \mathbf{c}'^T \Sigma \mathbf{c}')} \right)$ is the neural network kernel [Rasmussen, 2003].

Since the range of k_{NN} is $[-1, 1]$, f_{NN} is positive and thus satisfies **FM-P1**. Through Eq.7, Eq.2 is positive definite (Supp. Sec.3, Prop.2) and thus property **FM-P2** is satisfied. If the premise $t_1 < t_2$ of the property **FM-P3** is replaced by $t_1 > t_2$, then **FM-P3** is also satisfied. In contrast to the frequency modulation principle with distances in Eq. (3), the frequency modulation principle with a kernel is formalized as

$$
\begin{aligned}
& k_{NN}(\mathbf{c}, \mathbf{c}' | \Sigma) \geq k_{NN}(\tilde{\mathbf{c}}, \tilde{\mathbf{c}}' | \Sigma) \\
& \Rightarrow k((\mathbf{c}, v), (\mathbf{c}', v')) \geq k((\tilde{\mathbf{c}}, v), (\tilde{\mathbf{c}}', v'))
\end{aligned} \tag{8}
$$

Note that $k_{NN}(\mathbf{c}, \mathbf{c}' | \Sigma)$ is a similarity measure and thus the inequality is not reversed unlike Eq. (3).

All above arguments on the extension of the frequency modulation using a nonstationary kernel hold also when the k_{NN} is replaced by an arbitrary positive definite kernel. The only required condition is that a kernel has to be upper bounded, *i.e.*, $k_{NN}(\mathbf{c}, \mathbf{c}') \leq C$, needed for **FM-P1** and **FM-P2**.

4 RELATED WORK

On continuous variables, many sophisticated kernels have been proposed [Wilson and Nickisch, 2015, Samo and Roberts, 2015, Remes et al., 2017, Oh et al., 2018]. In contrast, kernels on discrete variables have been studied less [Haussler, 1999, Kondor and Lafferty, 2002, Smola and Kondor, 2003]. To our best knowledge, most of existing kernels on mixed variables are constructed by a kernel product Swersky et al. [2013], Li et al. [2016] with some exceptions [Krause and Ong, 2011, Swersky et al., 2013, Fiducioso et al., 2019], which rely on kernel addition.

In mixed variable BO, non-GP surrogate models are more prevalent, including SMAC [Hutter et al., 2011] using random forest and TPE [Bergstra et al., 2011] using a tree structured density estimator. Recently, by extending the approach of using Bayesian linear regression for discrete variables [Baptista and Poloczek, 2018], Daxberger et al. [2019] proposes Bayesian linear regression with manually chosen basis functions on mixed variables, providing a regret analysis using Thompson sampling as an acquisition function. Another family of approaches utilizes a bandit framework to handle the acquisition function optimization on mixed variables with theoretical analysis [Gopakumar et al., 2018, Nguyen et al., 2019, Ru et al., 2020]. Nguyen et al. [2019] use GP in combination with multi-armed bandit to model category-specific continuous variables and provide regret analysis using GP-UCB. Among these approaches, Ru et al. [2020] also utilize information across different categorical values, which –in combination with the bandit framework– makes itself the most competitive method in the family.

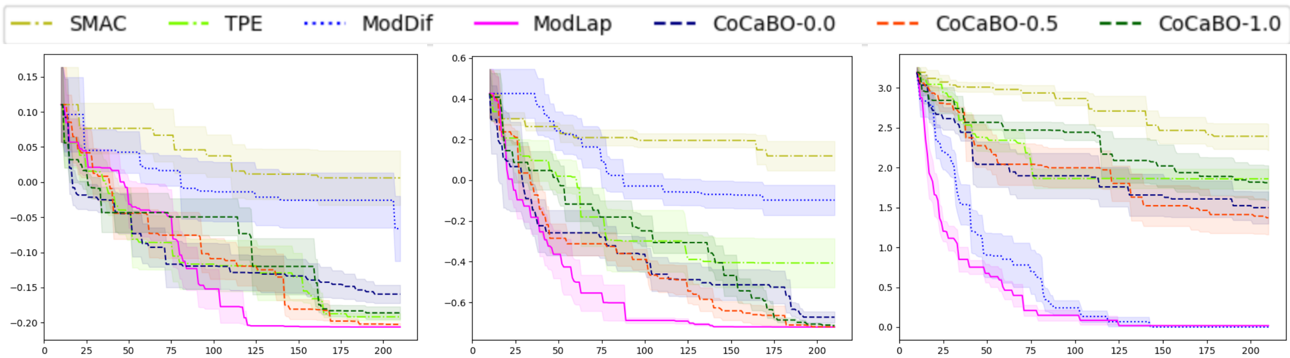
Our focus is to extend the modelling prowess and flexibility of pure GPs for surrogate models on problems with mixed variables. We propose frequency modulated kernels, which are kernels that are specifically designed to model the complex interactions between continuous and discrete variables.

In architecture search, approaches using weight sharing such as DARTS [Liu et al., 2018] and ENAS [Pham et al., 2018] are gaining popularity. In spite of their efficiency, methods training neural networks from scratch for given architectures outperform approaches based on weight sharing [Dong et al., 2021]. Moreover, the joint optimization of learning hyperparameters and architectures is under-explored with a few exceptions such as BOHB [Falkner et al., 2018] and autoHAS [Dong et al., 2020]. Our approach proposes a competitive option to this challenging optimization of mixed variable functions with expensive evaluation cost.

5 EXPERIMENTS

To demonstrate the improved sample efficiency of GP BO using FM kernels (BO-FM) we study various mixed variable black-box function optimization tasks, including 3 synthetic

Preliminary version



	SMAC	TPE	ModDif	ModLap	CoCaBO-0.0	CoCaBO-0.5	CoCaBO-1.0
Func2C	$+0.006 \pm 0.039$	-0.192 ± 0.005	-0.066 ± 0.046	$\mathbf{-0.206 \pm 0.000}$	-0.159 ± 0.013	$\mathit{-0.202 \pm 0.002}$	-0.186 ± 0.009
Func3C	$+0.119 \pm 0.072$	-0.407 ± 0.120	-0.098 ± 0.074	$\mathbf{-0.722 \pm 0.000}$	-0.673 ± 0.027	$\mathit{-0.720 \pm 0.002}$	-0.714 ± 0.005
Ackley5C	$+2.381 \pm 0.165$	$+1.860 \pm 0.125$	$\mathbf{+0.001 \pm 0.000}$	$\mathit{+0.019 \pm 0.006}$	$+1.499 \pm 0.201$	$+1.372 \pm 0.211$	$+1.811 \pm 0.217$

Figure 2: Func2C(left), Func3C(middle), Ackley5C(right) (Mean±Std.Err. of 5 runs)

problems from Ru et al. [2020], 2 hyperparameter optimization problems (SVM [Smola and Kondor, 2003] and XGBoost [Chen and Guestrin, 2016]) and the joint optimization of neural architecture and SGD hyperparameters.

As per our method, we consider ModLap which is of the form Eq. 5 with the following frequency modulating function.

$$f_{Lap}(\lambda, \| \mathbf{c} - \mathbf{c}' \|_\theta | \alpha, \beta) = \frac{1}{1 + \beta\lambda + \alpha \| \mathbf{c} - \mathbf{c}' \|_\theta^2} \tag{9}$$

Moreover, to empirically demonstrate the importance of the similarity measure behavior, we consider another kernel following the form of Eq. 5 but disrespecting the frequency modulation principle with the function

$$f_{Dif}(\lambda, \| \mathbf{c} - \mathbf{c}' \|_\theta | \alpha, \beta) = \exp\left(-(1 + \alpha \| \mathbf{c} - \mathbf{c}' \|_\theta^2)\beta\lambda\right) \tag{10}$$

We call the kernel constructed with this function ModDif. The implementation of these kernels is publicly available.[4]

In each round, after updating with an evaluation, we fit a GP surrogate model using marginal likelihood maximization with 10 random initialization until convergence [Rasmussen, 2003]. We use the expected improvement (EI) acquisition function [Donald, 1998] and optimize it by repeated alternation of L-BFGS-B [Zhu et al., 1997] and hill climbing [Skiena, 1998] until convergence. More details on the experiments are provided in Supp. Sec. 4.

Baselines For synthetic problems and hyperparameter optimization problems below, baselines we consider[5] are SMAC[6] [Hutter et al., 2011], TPE[7] [Bergstra et al., 2011], and CoCaBO[8] [Ru et al., 2020] which consistently outperforms One-hot BO [authors, 2016] and EXP3BO [Gopakumar et al., 2018]. For CoCaBO, we consider 3 variants using different mixture weights.[9]

5.1 SYNTHETIC PROBLEMS

We test on 3 synthetic problems proposed in Ru et al. [2020][10]. Each of the synthetic problems has the search space as in Tab. 1. Details of synthetic problems can be found in Ru et al. [2020].

	Conti. Space	Num. of Cats.
Func2C	$[-1,1]^2$	3, 5
Func3C	$[-1,1]^2$	3, 5,4
Ackley5C	$[-1,1]$	17, 17, 17, 17, 17

Table 1: Synthetic Problem Search Spaces

On all 3 synthetic benchmarks, ModLap shows competitive performance (Fig. 2). On Func2C and Func3C, ModLap performs the best, while on Ackley5C ModLap is at the second place, marginally further from the first. Notably, even on Func2C and Func3C, where ModDif underperforms significantly, ModLap exhibits its competitiveness, which empirically supports that the similarity measure behavior plays an important role in the surrogate modeling in Bayesian optimization. Note that TPE and CoCaBO have much shorter wall-clock runtime.

5.2 HYPERPARAMETER OPTIMIZATION PROBLEMS

Now we consider a practical application of Bayesian optimization over mixed variables. We take two machine learn-

[4] https://github.com/ChangYong-Oh/FrequencyModulatedKernelBO

[5] The methods [Daxberger et al., 2019, Nguyen et al., 2019] whose code has not been released are excluded.

[6] https://github.com/automl/SMAC3

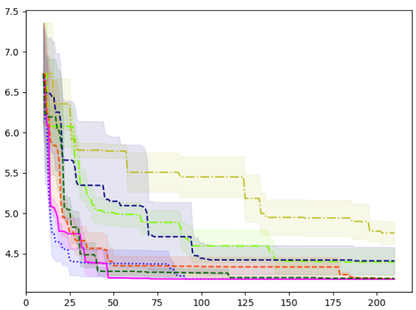
[7] http://hyperopt.github.io/hyperopt/

[8] https://github.com/rubinxin/CoCaBO_code

[9] Learning the mixture weight is not supported in the implementation, we did not include it. Moreover, as shown in Ru et al. [2020], at least one of 3 variants usually performs better than learning the mixture weight.

[10] In the implementation provided by the authors, only Func2C and Func3C are supported. We implemented Ackley5C.

Preliminary version



SVM	Method	XGBoost
$4.759 \pm .141$	SMAC	$.1215 \pm .0045$
$4.399 \pm .163$	TPE	$.1084 \pm .0007$
$4.188 \pm .001$	ModDif	*$.1071 \pm .0013$*
$4.186 \pm .002$	ModLap	**$.1038 \pm .0003$**
$4.412 \pm .170$	CoCaBO-0.0	$.1184 \pm .0062$
$4.196 \pm .004$	CoCaBO-0.5	$.1079 \pm .0010$
$4.196 \pm .004$	CoCaBO-1.0	$.1086 \pm .0008$

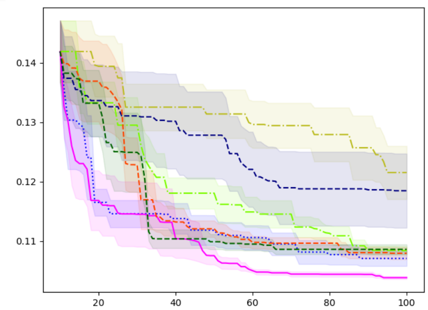


Figure 3: SVM(left), XGBoost(right) (Mean±Std.Err. of 5 runs)

ing algorithms, SVM [Smola and Kondor, 2003] and XGBoost [Chen and Guestrin, 2016] and optimize their hyperparameters.

SVM We optimize hyperparameters of NuSVR in scikit-learn [Pedregosa et al., 2011]. We consider 3 categorical hyperparameters and 3 continuous hyperparameters (Tab. 2) and for continuous hyperparameters we search over $\log_{10}$ transformed space of the range.

NuSVR param.[11]	Range
kernel	{linear, poly, RBF, sigmoid }
gamma	{scale, auto }
shrinking	{on, off }
C	$[10^{-4}, 10]$
tol	$[10^{-6}, 1]$
nu	$[10^{-6}, 1]$

Table 2: NuSVR hyperparameters

For each of 5 split of Boston housing dataset with train:test(7:3) ratio, NuSVR is fitted on the train set and RMSE on the test set is computed. The average of 5 test RMSE is the objective.

XGBoost We consider 1 ordinal, 3 categorical and 4 continuous hyperparameters (Tab. 3).

XGBoost param.[12]	Range
max_depth	$\{1, \cdots, 10\}$
booster	{gbtree, dart}
grow_policy	{depthwise, lossguide}
objective	{multi:softmax, multi:softprob}
eta	$[10^{-6}, 1]$
gamma	$[10^{-4}, 10]$
subsample	$[10^{-3}, 1]$
lambda	$[0, 5]$

Table 3: XGBoost hyperparameters

For 3 continuous hyperparameters, eta, gamma and subsample, we search over the $\log_{10}$ transformed space of the range.

[11] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVR.html`

[12] `https://xgboost.readthedocs.io/en/latest/parameter.html`

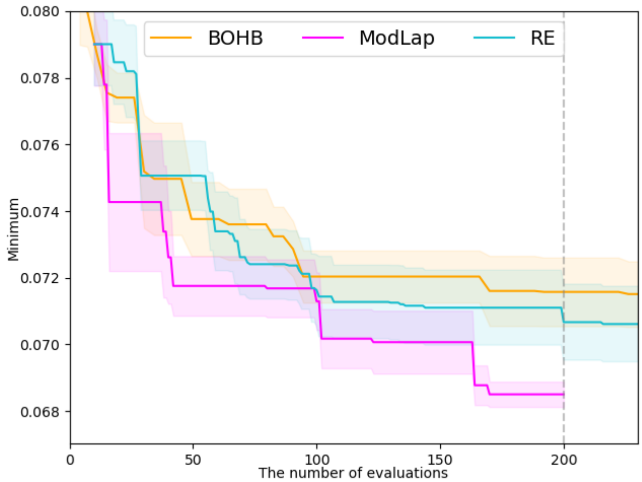
With a stratified train:test(7:3) split, the model is trained with 50 rounds and the best test error over 50 rounds is the objective of SVM hyperparameter optimization.

In Fig. 3, ModLap performs the best. On XGBoost hyperparameter optimization, ModLap exhibits clear benefit compared to the baselines. Here, ModDif wins the second place in both problems.

Comparison to different kernel combinations In Supp. Sec. 5, we also report the comparison with different kernel combinations on all 3 synthetic problems and 2 hyperparameter parameter optimization problems. We make two observations. First, ModDif, which does not respect the similarity measure behavior, sometimes severely degrades BO performance. Second, ModLap obtains equally good final results and consistently finds the better solutions faster than the kernel product. This can be clearly shown by comparing the area above the mean curve of BO runs using different kernels. The area above the mean curve of BO using ModLap is larger than the are above the mean curve of BO using the kernel product. Moreover, the gap between the area from ModLap and the area from kernel product increases in problems with larger search spaces. Even on the smallest search space, Func2C, ModLap lags behind the kernel product up to around 90th evaluation and outperforms after it. The benefit of ModLap modeling complex dependency among mixed variables is more prominent in higher dimension problems.

Ablation study on regression tasks In addition to the results on BO experiments, we compare FM kernels with kernel addiition and kernel product on three regression tasks from UCI datasets (Supp. Sec. 6). In terms of negative log-likelihood (NLL), which takes into account uncertainty, ModLap performs the best in two out of three tasks. Even on the task which is conjectured to have a structure suitable to kernel product, ModLap shows competitive performance. Moreover, on regression tasks, the importance of the frequency modulation principle is further reinforced. For full NLL and RMSE comparison and detailed discussion, see Supp. Sec. 6

Preliminary version



Method	#Eval.	Mean±Std.Err.
BOHB	200	$7.158 \times 10^{-2} \pm 1.0303 \times 10^{-3}$
BOHB	230	$7.151 \times 10^{-2} \pm 9.8367 \times 10^{-4}$
BOHB	600	$6.941 \times 10^{-2} \pm 4.4320 \times 10^{-4}$
RE	200	$7.067 \times 10^{-2} \pm 1.1417 \times 10^{-3}$
RE	230	$7.061 \times 10^{-2} \pm 1.1329 \times 10^{-3}$
RE	400	$6.929 \times 10^{-2} \pm 6.4804 \times 10^{-4}$
RE	600	$6.879 \times 10^{-2} \pm 1.0039 \times 10^{-3}$
ModLap	200	$\mathbf{6.850 \times 10^{-2}} \pm \mathbf{3.7914 \times 10^{-4}}$

For the figure with all numbers above, see Supp. Sec. 5.

Figure 4: Joint optimization of the architecture and SGD hyperparameters (Mean±Std.Err. of 4 runs)

5.3 JOINT OPTIMIZATION OF NEURAL ARCHITECTURE AND SGD HYPERPARAMETERS

Next, we experiment with BO on mixed variables by optimizing continuous and discrete hyperparameters of neural networks. The space of discrete hyperparameters $\mathscr{A}$ is modified from the NASNet search space [Zoph and Le, 2016], which consists of 8,153,726,976 choices. The space of continuous hyperparameters $\mathscr{H}$ comprises 6 continuous hyperparameters of the SGD with a learning rate scheduler: learning rate, momentum, weight decay, learning rate reduction factor, 1st reduction point ratio and 2nd reduction point ratio. A good neural architecture should both achieve low errors and be computationally modest. Thus, we optimize the objective $f(a,h) = err_{valid}(a,h) + 0.02 \times FLOP(a)/\max_{a' \in \mathscr{A}} FLOP(a')$. To increase the separability among smaller values, we use $\log f(a,h)$ transformed values whenever model fitting is performed on evaluation data. The reported results are still the original non-transformed $f(a,h)$.

We compare with two strong baselines. One is BOHB [Falkner et al., 2018] which is an evaluation-cost-aware algorithm augmenting unstructured bandit approach [Li et al., 2017] with model-based guidance. Another is RE [Real et al., 2019] based on a genetic algorithm with a novel population selection strategy. In Dong et al. [2021], on discrete-only spaces, these two outperform competitors including weight sharing approaches such as DARTS [Liu et al., 2018], SETN [Dong and Yang, 2019], ENAS [Pham et al., 2018] and etc. In the experiment, for BOHB, we use the public implementation[13] and for RE, we use our own implementation.

For a given set of hyperparameters, with ModLap or RE, the neural network is trained on FashionMNIST for 25 epochs while BOHB adaptively chooses the number of epochs. For further details on the setup and the baselines we refer the reader to Supp. Sec. 4 and 5.

We present the results in Fig. 4. Since BOHB adaptively chooses the budget (the number of epochs), BOHB is plotted according to the budget consumption. For example, the y-axis value of BOHB on 100-th evaluation is the result of BOHB having consumed 2,500 epochs (25 epochs × 100).

We observe that ModLap finds the best architecture in terms of accuracy and computational cost. What is more, we observe that ModLap reaches the better solutions faster in terms of numbers of evaluations. Even though the time to evaluate a new hyperparameter is dominant, the time to suggest a new hyperparameter in ModLap is not negligible in this case. Therefore, we also provide the comparison with respect the wall-clock time. It is estimated that RE and BOHB evaluate 230 hyperparameters while ModLap evaluate 200 hyperparameters (Supp. Sec. 4). For the same estimated wall-clock time, ModLap(200) outperforms competitors(RE(230), BOHB(230)).

In order to see how beneficial the sample efficiency of BO-FM is in comparison to the baselines, we perform a stress test in which more evaluations are allowed for RE and BOHB. We leave RE and BOHB for 600 evaluations. Notably, ModLap with 200 evaluations outperforms both competitors with 600 evaluations (Fig. 4 and Supp.Sec. 5). We conclude that ModLap exhibits higher sample efficiency than the baselines.

6 CONCLUSION

We propose FM kernels to improve the sample efficiency of mixed variable Bayesian optimization.

On the theoretical side, we provide and prove conditions for FM kernels to be positive definite and to satisfy the similarity measure behavior. Both conditions are not trivial due to the interactions between quantities on two disparate

[13] https://github.com/automl/HpBandSter

Preliminary version

domains, the spatial domain and the frequency domain.

On the empirical side, we validate the effect of the conditions for FM kernels on multiple synthetic problems and realistic hyperparameter optimization problems. Further, we successfully demonstrate the benefits of FM kernels compared to non-GP based Bayesian Optimization on a challenging joint optimization of neural architectures and SGD hyperparameters. BO-FM outperforms its competitors, including Regularized evolution, which requires three times as many evaluations.

We conclude that an effective modeling of dependencies between different types of variables improves the sample efficiency of BO. We believe the generality of the approach can have a wider impact on modeling dependencies between discrete variables and variables of arbitrary other types, including continuous variables.

References

The GPyOpt authors. Gpyopt: A bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`, 2016.

Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *ICML*, pages 462–471, 2018.

James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *NeurIPS*, pages 2546–2554, 2011.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas. Bayesian optimization in alphago. *arXiv preprint arXiv:1812.06855*, 2018.

Erik Daxberger, Anastasia Makarova, Matteo Turchetta, and Andreas Krause. Mixed-variable bayesian optimization. *arXiv preprint arXiv:1907.01329*, 2019.

R Jones Donald. Efficient global optimization of expensive black-box function. *J. Global Optim.*, 13:455–492, 1998.

Xuanyi Dong and Yi Yang. One-shot neural architecture search via self-evaluated template network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3681–3690, 2019.

Xuanyi Dong, Mingxing Tan, Adams Wei Yu, Daiyi Peng, Bogdan Gabrys, and Quoc V Le. Autohas: Efficient hyperparameter and architecture search. *arXiv preprint arXiv:2006.03656*, 2020.

Xuanyi Dong, Lu Liu, Katarzyna Musial, and Bogdan Gabrys. Nats-bench: Benchmarking nas algorithms for architecture topology and size. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, pages 1437–1446. PMLR, 2018.

Marcello Fiducioso, Sebastian Curi, Benedikt Schumacher, Markus Gwerder, and Andreas Krause. Safe contextual bayesian optimization for sustainable room temperature pid control tuning. In *IJCAI*, pages 5850–5856. AAAI Press, 2019.

Shivapratap Gopakumar, Sunil Gupta, Santu Rana, Vu Nguyen, and Svetha Venkatesh. Algorithmic assurance: An active approach to algorithmic testing using bayesian optimisation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5470–5478, 2018.

David Haussler. Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California . . . , 1999.

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

Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Neural architecture search with bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems*, pages 2016–2025, 2018.

Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *ICML*, 2002.

Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabas Poczos, Jeff Schneider, and Eric P Xing. Chembo: Bayesian optimization of small organic molecules with synthesizable recommendations. *arXiv preprint arXiv:1908.01425*, 2019.

Andreas Krause and Cheng S Ong. Contextual gaussian process bandit optimization. In *NeurIPS*, pages 2447–2455, 2011.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

Preliminary Version

Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *ICML*, volume 16, pages 1245–1253, 2016.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Dang Nguyen, Sunil Gupta, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Bayesian optimization for categorical and category-specific continuous inputs. *arXiv preprint arXiv:1911.12473*, 2019.

ChangYong Oh, Efstratios Gavves, and Max Welling. Bock: Bayesian optimization with cylindrical kernels. In *ICML*, pages 3868–3877, 2018.

Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial bayesian optimization using the graph cartesian product. In *NeurIPS*, pages 2910–2920, 2019.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*, pages 4095–4104. PMLR, 2018.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019.

Sami Remes, Markus Heinonen, and Samuel Kaski. Non-stationary spectral kernels. In *NeurIPS*, pages 4642–4651, 2017.

Jean-Louis Reymond and Mahendra Awale. Exploring chemical space for drug discovery using the chemical universe database. *ACS chemical neuroscience*, 3(9):649–657, 2012.

Binxin Ru, Ahsan Alvi, Vu Nguyen, Michael A Osborne, and Stephen Roberts. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning*, pages 8276–8285. PMLR, 2020.

Yves-Laurent Kom Samo and Stephen Roberts. Generalized spectral kernels. *arXiv preprint arXiv:1506.02236*, 2015.

Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Steven S Skiena. *The algorithm design manual: Text*, volume 1. Springer Science & Business Media, 1998.

Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *NeurIPS*, pages 2951–2959, 2012.

Benjamin Solnik, Daniel Golovin, Greg Kochanski, John Elliot Karro, Subhodeep Moitra, and D. Sculley. Bayesian optimization for a better dessert. In *Proceedings of the 2017 NIPS Workshop on Bayesian Optimization*, December 9, 2017, Long Beach, USA, 2017. The workshop is BayesOpt 2017 NIPS Workshop on Bayesian Optimization December 9, 2017, Long Beach, USA.

Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task bayesian optimization. In *NeurIPS*, pages 2004–2012, 2013.

Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. A primer on kernel methods. *Kernel methods in computational biology*, 47:35–70, 2004.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *ICML*, pages 1775–1784, 2015.

Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

Preliminary version