Bayesian Optimisation for Mixed-Variable Inputs using Value Proposals

Yan Zuo [1 2]   **Amir Dezfouli** [2]   **Iadine Chadès** [2]   **David Alexander** [2]   **Benjamin Ward Muir** [1]

Abstract

Many real-world optimisation problems are defined over both categorical and continuous variables, yet efficient optimisation methods such as Bayesian Optimisation (BO) are not designed to handle such mixed-variable search spaces. Recent approaches to this problem cast the selection of the categorical variables as a bandit problem, operating independently alongside a BO component which optimises the continuous variables. In this paper, we adopt a holistic view and aim to consolidate optimisation of the categorical and continuous sub-spaces under a single acquisition metric. We derive candidates from the Expected Improvement criterion, which we call value proposals, and use these proposals to make selections on both the categorical and continuous components of the input. We show that this unified approach significantly outperforms existing mixed-variable optimisation approaches across several mixed-variable black-box optimisation tasks.

1. Introduction

Bayesian optimisation (BO) has established itself as an efficient method for optimising black-box functions that are costly to evaluate (Shahriari et al., 2015). Typical BO methods model the black-box function of interest using a surrogate statistical model (usually a Gaussian Process (GP) (Dudley, 2010)), seeking out the next point to evaluate by optimising a more tractable (typically differentiable) function called an acquisition function. The role of the acquisition function is to balance two conflicting requirements: exploitation of the current knowledge about the objective function and exploration to gain more knowledge about the objective function. BO has been applied effectively to tasks that range from experimental design (Griffiths & Hernández-Lobato, 2017; Shields et al., 2021) to hyperparameter search (Gardner et al., 2014; Snoek et al., 2012) in machine learning models. Notably, it is often observed that

BO is particularly well-suited for applications where the number of allowable evaluations on the objective function is limited (Bull, 2011).

However, many real world optimisation problems involve a mixture of continuous and categorical variables in the input space. For example, in automated machine learning applications (Hutter et al., 2019) where the aim is to automatically select a machine learning model along with its corresponding optimal hyperparameters, each model can be seen as a categorical choice while the hyperparameters of the model can be viewed as category-specific continuous variables. Another example is in the chemical reaction space, where often the function we are interested in optimising is represented by both categorical (compositional) variables and continuous (process) variables (Zhou et al., 2017). These scenarios present an additional challenge for current BO models (particularly those using GPs as their underlying surrogate models) which are ill-equipped to deal effectively with such problems containing multi-layered and complicated search spaces.

There are numerous challenges associated with optimising mixed-variable functions within a BO framework. In particular, the assumption that the acquisition function is differentiable over the input space (allowing for efficient optimisation) becomes invalid. Recent works have dealt with the categorical part of the input in various ways ranging from one-hot encoding of the categorical components (Golovin et al., 2017; Snoek et al., 2012) to using hierarchical models (Bergstra et al., 2011; Hutter et al., 2011) suited towards both categorical and continuous inputs. Several modern approaches for optimising mixed-variable functions utilise a mixed approach involving Multi-Armed Bandits (MABs) (Auer et al., 2002) and BO for handling the categorical and continuous parts respectively. However, these methods either treat the mixed-variable optimisation problem as a collection of smaller continuous problems (Gopakumar et al., 2018; Nguyen et al., 2020) which is sample-inefficient, or separately optimise for each categorical variable (Ru et al., 2020) which is a non-global approach.

In this paper, we present a new Bayesian Optimisation approach for optimising black-box functions with multiple continuous and categorical inputs. For carrying out a decision on the categorical part of the input, our method uses a

set of acquisition values derived from our surrogate model, which we call *value proposals*. This enables a global, unified approach for optimising on the discrete and continuous sub-spaces of the input, where the decision-making process for both categorical and continuous variables is based on a common metric obtained from the underlying surrogate. Bayesian Optimisation using Value Proposals (VPBO) offers the following main contributions:

- We propose a novel BO approach for optimising mixed-variable black-box functions using value proposals, unifying the joint optimisation of categorical and continuous inputs under a single framework.

- We derive the regret bound for VPBO, showing that the regret for our method grows only sub-linearly in the number of iterations.

- We show our method significantly improves over existing baselines when applied to a variety of mixed input synthetic and real-world problems.

2. Background

Generally, we can organise the literature related to our work into three categories:

One-hot Encoding Prior to the introduction of hierarchical and categorical-specific models for mixed input tasks, one-hot encoding methods (Golovin et al., 2017; González et al., 2016; Snoek et al., 2012), which transform categorical variables into a one-hot encoded representation, were used by Bayesian Optimisation frameworks to deal with inputs of mixed nature. In this scenario, the categorical variable with N choices is transformed into a vector of length N with a single non-zero element. Since categories are mutually exclusive, this type of approach treats each extra variable as continuous in $[0, 1]$ using a standard Bayesian Optimisation algorithm for optimisation. However, this type of approach places an equal measure of covariance between all category pairs (despite some or all pairs having different or no correlations), resulting in an acquisition function that is difficult to optimise with large areas of flatness (Rana et al., 2017). To address this issue, (Garrido-Merchán & Hernández-Lobato, 2020) restricted the objective function to change only at designated points of 0 and 1, using a kernel function which computed covariances after rounding off the input. However, with this approach, the resulting acquisition function becomes step-wise, making it difficult to optimise.

Hierarchical Another approach to dealing with mixed inputs is to change the underlying surrogate model for the black-box function from the commonly used Gaussian Process to a model that can more naturally consider both continuous and categorical variables. Sequential Model-based Algorithm Configuration (SMAC) (Hutter et al., 2011) uses Random Forests (RFs) (Breiman, 2001) as a surrogate model to handle both categorical and continuous components in the input. However, the random nature of RFs (through a reliance on bootstrapping samples and randomly choosing subsets of variables to be tested at each node) weakens the reliability of the derived acquisition function. Adding to this, RFs have a tendency to overfit to training data, requiring careful selection of the number of trees in the model to avoid overfitting. Other tree-based approaches include the Tree-Parzen Estimator (Bergstra et al., 2011), which uses tree-structured Parzen density estimators.

Category-specific The category-specific approach is to handle each component of the input separately. (Gopakumar et al., 2018) developed EXP3BO, an approach to deal with mixed categorical and continuous input spaces by utilising a Multi-Armed Bandit to make categorical choices and training a separate surrogate model specific for each choice of category. As a result, the observed data is divided into smaller subsets (one for each category), resulting in a sample-inefficient optimisation procedure that cannot handle problems with a large number of categorical choices. Nguyen et al. (2020) introduced a batched setting to the optimisation framework of Gopakumar et al. (2018) and replaced the respective EXP3 and Upper Confidence Bound (UCB) algorithms of the framework with Thompson Sampling (Chapelle & Li, 2011). A key limitation of these frameworks is that they only allow for optimisation of a single categorical variable; the work of Ru et al. (2020) extended this type of approach by allocating a MAB per categorical variable, enabling the optimisation of functions with multiple categorical variables. However, each MAB is individually updated using the EXP3 algorithm and this may lead to estimates which are disjoint from the BO-backend. Our method builds on these hybrid approaches but instead adopts a global view for selecting categorical combinations in the discrete sub-space of the input.

3. Preliminaries

3.1. Problem Setup

We consider the problem of optimising a black-box function $f(\boldsymbol{z})$ where the input $\boldsymbol{z}$ is comprised of categorical and continuous parts *i.e.* $\boldsymbol{z} = [\boldsymbol{h}, \boldsymbol{x}]$. Here, $\boldsymbol{h} = [h_1, ..., h_k]$ is a vector of categorical variables from the discrete topological space $\mathcal{C}$, with each categorical variable $h_i \in \{1, 2, ..., N_j\}$ taking one of N_j different values. The continuous component of the input, $\boldsymbol{x}$, is drawn from a d_x-dimensional hypercube $\mathcal{X}$. Formally, the optimisation of the black-box function f is expressed as:

$$\boldsymbol{z}^* = [\boldsymbol{h}^*, \boldsymbol{x}^*] = \underset{z}{\operatorname{argmax}} f(\boldsymbol{z}) \qquad (1)$$

which is performed sequentially by making a series of evaluations on $z_1, ... z_T$. The goal is to find the best configuration $\boldsymbol{z}^*$ that maximises y, which is the value returned from our objective function f. For convenience, we use c to denote a possible combination of categorical choices out of $C = \prod_{j=1}^{k} N_j$ available combinations, such that $c \in \{1, ..., C\}$.

3.2. Bayesian Optimisation

Given a black-box objective function $f : \mathcal{X} \to \mathbb{R}$, the goal of BO is to find the optimal value $\boldsymbol{x}^*$ under a setting with limited evaluations on f. This optimal value maximises the objective f, and is defined as $\boldsymbol{x}^* = \mathrm{argmax}_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$. The process of BO involves using a surrogate to model the objective f; typically, f is assumed to be a smooth function and commonly a Gaussian Process (GP) is used for the surrogate. The GP models an underlying probability distribution over functions f and is represented by mean and covariance functions (or kernel) $\mu(\boldsymbol{x})$ and $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ respectively, where $f(\boldsymbol{x}) \sim \mathrm{GP}(\mu(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}'))$.

The surrogate encodes our prior beliefs about the objective f and we can build a posterior through further observations when we evaluate $f(\boldsymbol{x})$. Using this posterior, at a given optimisation iteration t, an acquisition function $\alpha_t(\boldsymbol{x})$ can be built, which can then be optimised to identify the next point to be sampled such that $\boldsymbol{x}_t = \mathrm{argmax}_{x \in \mathcal{X}} \alpha_t(\boldsymbol{x})$. Since $\alpha_t(\boldsymbol{x})$ is derived from our surrogate model, it is comparatively cheaper to compute and can be optimised using standard optimisation techniques.

3.3. Expected Improvement

For our choice of acquisition function, we use Expected Improvement (EI), which is an expectation over the improvement function. The improvement function at iteration t is given by:

$$\mathcal{I}_t(\boldsymbol{x}_t) = \max \{0, f(\boldsymbol{x}_t) - \mathcal{E}_t\}, \tag{2}$$

where $\mathcal{E}_t$ defines an incumbent at iteration t, such that $\mathcal{E}_t = y_{t-1}^{max} = \max \{y_1, ... y_{t-1}\}$. We denote $\gamma = \gamma_{t-1}(\boldsymbol{x}) = \frac{\mu_{t-1}(\boldsymbol{x}) - \mathcal{E}_t}{\sigma_{t-1}(\boldsymbol{x})}$, where $\mu_{t-1}(\boldsymbol{x})$ and $\sigma_{t-1}^2(\boldsymbol{x})$ are respectively the predictive mean and variance of the posterior. Then we can obtain the closed-form acquisition function at iteration t by taking the expectation over Eq. 14 (refer to Appendix A for derivation):

$$\alpha_t^{EI}(\boldsymbol{x}) = \mathbb{E}[\mathcal{I}_t(\boldsymbol{x})]$$
$$= \sigma_{t-1}(\boldsymbol{x})\phi(\gamma) + (\mu_{t-1}(\boldsymbol{x}) - \mathcal{E})\Phi(\gamma), \tag{3}$$

where ϕ and Φ are the PDF and CDF of the standard normal distribution respectively. In the case $\sigma_{t-1}(\boldsymbol{x}) = 0$, we define $\alpha_t^{EI}(\boldsymbol{x}) = 0$.

3.4. Mixed-Kernels

Under the typical setting of BO, where the surrogate is defined by a Gaussian Process, the GP kernel is defined over a continuous space. This makes using GP-based BO approaches on mixed-variable problems especially difficult. We adopt the mixed-kernel approach of Ru et al. (2020) which defines two separate kernels: a categorical kernel $\kappa_h(\boldsymbol{h}, \boldsymbol{h}')$ defined over the discrete sub-space and a continuous kernel $\kappa_x(\boldsymbol{x}, \boldsymbol{x}')$ over the continuous sub-space. The mixed-kernel $\kappa_z(\boldsymbol{z}, \boldsymbol{z}')$ is then defined as a mixture of sum and product kernels over κ_h and κ_x:

$$\kappa_z(\boldsymbol{z}, \boldsymbol{z}') = (1 - \lambda)(\kappa_h(\boldsymbol{h}, \boldsymbol{h}') + \kappa_x(\boldsymbol{x}, \boldsymbol{x}'))$$
$$+ \lambda \kappa_h(\boldsymbol{h}, \boldsymbol{h}')\kappa_x(\boldsymbol{x}, \boldsymbol{x}'), \tag{4}$$

with $\lambda \in [0, 1]$ being a trade-off parameter for the sum and product kernels which can be optimised jointly alongside the GP hyperparameters. The mixed-kernel is able to learn covariances between observations made in the joint input space $\mathcal{Z} = \mathcal{C} \times \mathcal{X}$. Samples from different categories will have covariances which are dominated by κ_x and when there is partial overlap in categorical choices between samples, the covariance is determined by contributions from both κ_h and κ_x (for additional details on the mixed-kernel, refer to the Appendix B).

4. Method

4.1. Bayesian Optimisation using Value Proposals

Our proposed method, Bayesian Optimisation using Value Proposals (VPBO), extends hybrid MAB-BO approaches such as Gopakumar et al. (2018); Nguyen et al. (2020); Ru et al. (2020) to adopt a global view for optimising mixed-variable input spaces. This allows our method to retain the advantages of bandit approaches for discrete-type problems such as selecting categorical inputs.

The VPBO procedure first queries the BO back-end by maximising the acquisition function to make a selection on the continuous part of the input $\boldsymbol{x}_t$. This step is done for each of the C possible combinations of categories. For each combination of categories c, VPBO generates a *value proposal*, which is collected in the set of all value proposals $\mathcal{V}_t$ for the optimisation iteration t, along with their corresponding $\boldsymbol{x}_{t,c}^*$. Deciding the categorical choices $\boldsymbol{h}_t^*$ at iteration t is then formulated as a MAB problem (Bubeck et al., 2008; Vermorel & Mohri, 2005), where VPBO makes a selection on the categorical inputs $\boldsymbol{h}_t$ using the value proposal set $\mathcal{V}_t$. The maximum acquisition value for a proposal out of the proposal set $\mathcal{V}_t$ is selected and its corresponding input $[\boldsymbol{h}_t^*, \boldsymbol{x}_t^*]$ is chosen as the next query point of the objective function f. Following this, the newly observed function value f_t^* and corresponding input z_t^* is added to the observation set $\mathcal{D}_t$.

The result is that when selecting the value proposal, we

are able to choose from all the usual acquisition metrics (*e.g.* UCB, EI, *etc.*) and use this single metric to select both the categorical *and* continuous components of the input. The surrogate GP model is the singular driving mechanism behind our entire method, all whilst factoring in the uncertainty of the surrogate in the decision making process for the discrete and continuous sub-spaces of the input. This is in contrast to other hybrid MAB-BO frameworks which require separate standalone bandits in addition to a BO back-end (Nguyen et al., 2020; Ru et al., 2020).

In summary, our method is described as follows: at each iteration t, for each possible combination of categorical choices, we employ the GP-based BO back-end and optimise the continuous sub-space for each combination of categories, given our categorical variables. The resulting optimisation over the continuous sub-space produces a value proposal for each combination of categories and we make a selection on the decision variables in the discrete sub-space which maximises the proposal value over all possible categorical combinations. Our VPBO procedure is detailed in Algorithm 1.

4.2. Theoretical Analysis

We now show that our method converges to the global maximum with a sub-linear rate in the mixed space setting. At optimisation iteration t, VPBO performs the following:

$$
\begin{aligned}
[\boldsymbol{h}_t^*, \boldsymbol{x}_t^*] &= \underset{\boldsymbol{z}}{\operatorname{argmax}}\, \alpha_t(\boldsymbol{z}|\mathcal{D}_{t-1}) \\
&= \underset{\boldsymbol{h}}{\operatorname{argmax}}\, \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \alpha_t(\boldsymbol{x}|\mathcal{D}_{t-1}, \boldsymbol{h}).
\end{aligned} \quad (5)
$$

Algorithm 1 VPBO Optimisation

Input Black-box function f, Initial observation data $\mathcal{D}_0$, Maximum number of iterations T, Number of possible categorical combinations C
Output The best recommendation $\boldsymbol{z}_T^* = [\boldsymbol{h}_T^*, \boldsymbol{x}_T^*]$
1: Initialise the data $\mathcal{D}_0$
2: **for all** $t = \{1, ..., T\}$ **do**
3:     Fit GP using $\mathcal{D}_{t-1}$
4:     **for each** $c = \{1, ..., C\}$ **do**
5:         $\boldsymbol{x}_{t,c}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \alpha_{t,c}(\boldsymbol{x}|\mathcal{D}_{t-1}, \boldsymbol{h}_{t,c})$
6:         $v_{t,c} = \alpha_{t,c}(\boldsymbol{x}_{t,c}^*|\mathcal{D}_{t-1}, \boldsymbol{h}_{t,c})$
7:         $\mathcal{V}_t[c] = [\boldsymbol{x}_{t,c}^*, v_{t,c}]$ ▷ Add $[\boldsymbol{x}_{t,c}^*, v_{t,c}]$ to proposal set
8:     **end for**
9:     $c^* = \underset{c}{\operatorname{argmax}}\, \mathcal{V}_t$
10:    Set $\boldsymbol{z}_t^* = [\boldsymbol{h}_t^*, \boldsymbol{x}_t^*] = [\boldsymbol{h}_{t,c^*}, \boldsymbol{x}_{t,c^*}]$
11:    Query at $\boldsymbol{z}_t^*$ to obtain f_t^*
12:    Augment the data: $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \bigcup (\boldsymbol{z}_t^*, f_t^*)$
13: **end for**

This is equivalent to a Bayesian Optimisation setting over a mixed search space, where each continuous component of the mixed search space corresponds to a combination of categories c. We consider the noisy case $y_{c,t} = f_c(\boldsymbol{x}) + \xi_t$, where $\xi_t \sim \mathcal{N}(0, \sigma_\xi^2)$. To simplify our notation, when referencing in the context of Eq. 5, we drop the c term from $\boldsymbol{x}_c$ and f_c, using the notation $\boldsymbol{x} \in \mathcal{X}_c$ to resolve any ambiguity. Following Srinivas et al. (2009), we assume our objective f is smooth according to the Reproducing Kernel Hilbert Space (RKHS) associated with a GP with a mixed-kernel κ_z in the mixed search space setting:

Assumption 1. *The objective function f is a member of the Reproducing Kernel Hilbert Space $\mathcal{H}_{\kappa_z}([\mathcal{C}, \mathcal{X}])$ with known kernel function κ_z, and has a bounded norm $||f||_{\mathcal{H}_{\kappa_z}} \leq B$, for some $B > 0$.*

This assumption is central for Bayesian Optimisation in justifying the use of GPs to estimate f from samples (Rasmussen, 2003) and ensuring our corresponding regret bounds hold. To show the convergence of VPBO in the mixed search space setting, we assume f is bounded in $[\mathcal{C}, \mathcal{X}]$, which is common as it is generally assumed in BO that f is Lipschitz continuous (Brochu et al., 2010):

Assumption 2. *The objective function $f(\boldsymbol{z})$ is bounded in $[\mathcal{C}, \mathcal{X}]$, i.e. $\exists F_1, F_2 \in \mathbb{R} : \forall \boldsymbol{z} \in [\mathcal{C}, \mathcal{X}], F_1 \leq f(\boldsymbol{z}) \leq F_2$.*

Using Assumptions 1 and 2 and the assumptions from Vazquez & Bect (2010), we can show the optimisation in Eq. 5 can be treated as a general BO problem, with the surrogate defined as a GP with mixed-kernel κ_z and using Expected Improvement as our acquisition criterion α; under this setting, κ_z should be a valid kernel (Bull, 2011).

Lemma 1. *The mixed kernel κ_z is a valid kernel (i.e. positive semi-definite).*

Proof. The categorical kernel κ_h is demonstrated to be a valid kernel in Ru et al. (2020). The mixed kernel κ_z is additive and multiplicative between two valid kernels κ_h and κ_x which results in a valid kernel. Therefore, the mixed kernel κ_z is also a valid kernel. This concludes the proof. $\square$

We relax the continuous kernel assumption of Vazquez & Bect (2010) and derive the maximum information gain of our mixed-kernel κ_z. We define the *maximum information gain* complexity measure (Srinivas et al., 2009) as:

Definition 1. *Given a sequence of choices $A = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_T\} \subset \mathcal{Z}$, let $f_A = \{f(\boldsymbol{z}_i)\}$, $y_A = \{f(\boldsymbol{z}_i) + \xi_i\}$ and I be the mutual information. The maximum information gain ψ_T after T iterations is defined as:*

$$
\psi_T := \max_{A \subset \mathcal{Z}, |A| = T} I(y_A; f_A). \quad (6)
$$

We show that the information gain of the mixed-kernel after T iterations, ψ_T^z, is bounded by some maximum value. For

convenience, we denote the maximum information gain on the mixed kernel as $\psi_T^z := \psi_T(\kappa_z; [\mathcal{C}, \mathcal{X}])$, that on the categorical kernel as $\psi_T^h := \psi_T(\kappa_h; \mathcal{C})$ and that on the continuous kernel as $\psi_T^x := \psi_T(\kappa_x; \mathcal{X})$.

Lemma 2. *For the mixed-kernel κ_z, the maximum information gain ψ_T^z after T iterations is given by:*

$$\psi_T^z \leq \mathcal{O}\big((\lambda C + 1 - \lambda)(T^{1-2\eta}(\log T)) \\ + (C + 2 - 2\lambda)\log T\big), \qquad (7)$$

where $\eta = \frac{\nu}{2\nu + d_x(d_x+1)}$, $\nu > 2$ and $d_x \geq 1$ are the smoothness parameter and dimension of the Matérn kernel κ_x, C is the number of possible categorical combinations and $\lambda \in [0,1]$ is the trade-off parameter between the sum and product kernels of κ_h and κ_x.

Proof. From Krause & Ong (2011), given two kernels κ_h and κ_x, and if κ_h is a kernel on $\mathcal{C}$ with rank at most m, then the bound on the maximum information gain for the sum and product of these kernels is respectively given as:

$$\psi_T(\kappa_h + \kappa_x; [\mathcal{C}, \mathcal{X}]) \leq \psi_T^h + \psi_T^x + 2\log T \qquad (8)$$
$$\psi_T(\kappa_h \kappa_x; [\mathcal{C}, \mathcal{X}]) \leq m\psi_T^x + m\log T. \qquad (9)$$

Using Kirchhoff & Kuhnt (2020); Zhou et al. (2011), we can find the maximum rank r of κ_h by applying a Cholesky-type decomposition such that $r = \prod_{j=1}^{k} N_j = C$, where k is the number of categorical variables each with N_j distinct values. Applying this to Eq. 8 and 9, we have bounds on the information gain of the sum and product kernels for κ_h and κ_x respectively:

$$\psi_T(\kappa_h + \kappa_x; [\mathcal{C}, \mathcal{X}]) \leq \mathcal{O}(\psi_T^x + (C+2)\log T) \quad (10)$$
$$\psi_T(\kappa_h \kappa_x; [\mathcal{C}, \mathcal{X}]) \leq C\psi_T^x + C\log T. \quad (11)$$

The mixed-kernel κ_z is defined as $\kappa_z = \lambda \kappa_h \kappa_x + (1 - \lambda)(\kappa_h + \kappa_x)$ where $\lambda \in [0,1]$ (Ru et al., 2020). Using Eq. 10 and 11 with this definition, we obtain the following bound on the information gain of the mixed-kernel κ_z:

$$\psi_T^z \leq \lambda \mathcal{O}(C\psi_T^x + C\log T) \\ + (1-\lambda)(\psi_T^x + (C+2)\log T) \\ \leq \mathcal{O}((\lambda C + 1 - \lambda)\psi_T^x + (C+2-2\lambda)\log T). \quad (12)$$

From Srinivas et al. (2009), the maximum information gain on the continuous component of the mixed-kernel ψ_T^x, is given as: $\psi_T^x := \psi_T(\kappa_x; \mathcal{X}) \sim \mathcal{O}(T^{1-2\eta}(\log T))$, where $\eta = \frac{\nu}{2\nu + d_x(d_x+1)}$. Applying this to Eq. 12 yields Eq. 7, thus concluding the proof. $\qquad \square$

We now bound the regret at iteration T, $\mathcal{R}_T$, for our method:

Theorem 1. *Let σ_ξ^2 be the measurement noise variance, $\beta_T = 2\|f\|_{\kappa_z}^2 + 300\psi_T^z \log^3(T/\delta)$ and $\delta \in (0,1)$. Then, using $y^{max} = \max\limits_{y_i \in \mathcal{D}_t} y_i$ as the incumbent, the cumulative regret of EI with mixed-kernel κ_z is given by the sub-linear rate:*

$$\mathcal{R}_T \leq \sqrt{T\beta_T\psi_T^z} \sim \mathcal{O}\left(\sqrt{T^{1-\eta}(\log T)^3}\right), \qquad (13)$$

with probability at least $1 - \delta$.

Proof. Under the valid kernel setting of Lem. 1 and applying Lem. 2 to Theorem 4 from Nguyen et al. (2017), we obtain Eq. 13 and thus conclude the proof. $\qquad \square$

Therefore, the regret for our method increases at a sub-linear rate $\mathcal{R}_T \sim \mathcal{O}\left(\sqrt{T^{1-\eta}(\log T)^3}\right)$, which vanishes in the limit as $\lim_{T\to\infty} \frac{\mathcal{R}_T}{T} = 0$.

5. Experiments

We compared VPBO against several competing baselines which can handle mixed-variable type inputs: SMAC (Hutter et al., 2011), TPE (Bergstra et al., 2011), GP-based BO with one-hot encoding (One-Hot BO), EXP3BO (Gopakumar et al., 2018), Bandit-BO (Nguyen et al., 2020) and CoCaBO (Ru et al., 2020). All baseline methods are implemented according to their publicly available Python implementations[1]. Additionally, we implemented RandomBO, where the EXP3 bandit algorithm in CoCaBO is replaced by a bandit that selects categorical variables at random.

Competing Baseline Settings One-Hot BO, EXP3BO, Bandit-BO, RandomBO and CoCaBO use the Upper Confidence Bound (UCB) acquisition function (Srinivas et al., 2009) with trade-off parameter $k = 2.0$. For CoCaBO, we used the two best performing variations reported in the paper, using kernel mix values of $\lambda = \{0.5, \text{auto}\}$, where $\lambda = \text{auto}$ indicates λ being optimised as a hyperparameter. All experiments are run under the sequential setting (*i.e.* batch size of 1).

For SMAC, TPE, RandomBO and CoCaBO, we initialise each model with 24 randomly sampled initial points as was done in Ru et al. (2020). Both EXP3BO and Bandit-BO require more than 24 initial samples since these approaches need to fit a separate surrogate model for each categorical combination. That is, the initial observation data needs to be evenly divided and allocated to surrogate models for each

---

[1]One-Hot BO: https://github.com/SheffieldML/GPyOpt, SMAC: https://github.com/automl/SMAC3, TPE: https://github.com/hyperopt/hyperopt, EXP3BO: https://github.com/shivapratap/AlgorithmicAssurance_NIPS2018, Bandit-BO: https://github.com/nphdang/Bandit-BO, CoCaBO: https://github.com/rubinxin/CoCaBO_code
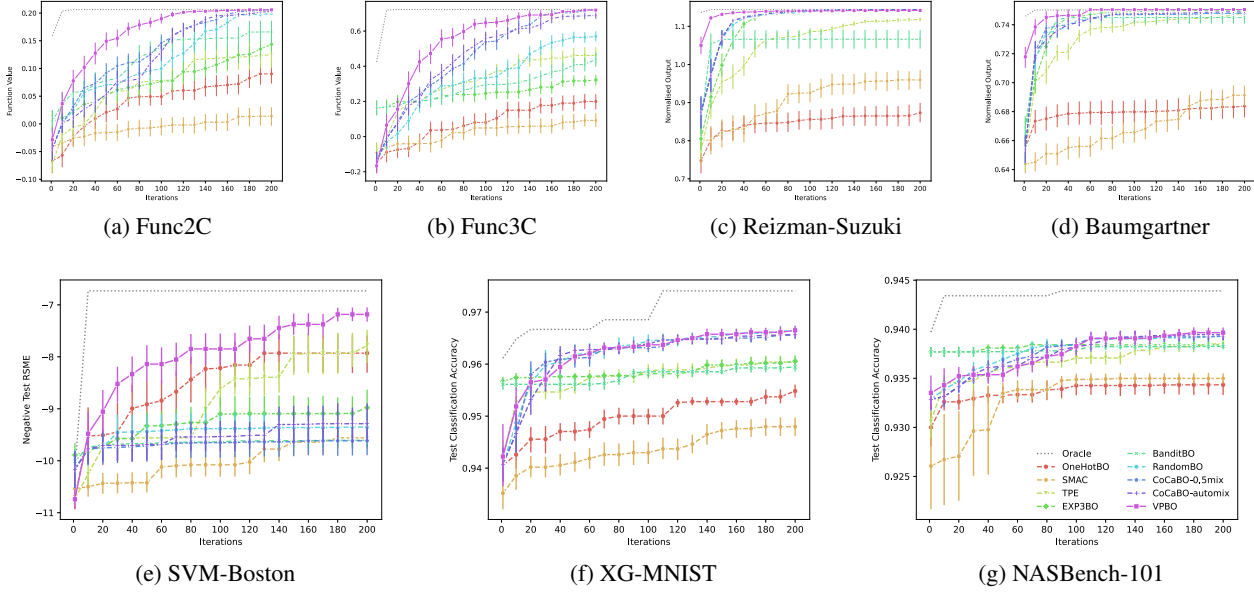
Figure 1. Performance of VPBO against existing methods on various synthetic and real-world tasks.

categorical combination. We follow the approach of Ru et al. (2020) and initialise each surrogate model with 3 randomly sampled initial points.

VPBO Settings   For our VPBO method, we utilise the same mixed-kernel GP surrogate as Ru et al. (2020), replacing the UCB acquisition function with the Expected Improvement (EI) acquisition function (Jones et al., 1998; Mockus et al., 1978).

In order to make VPBO computationally tractable for datasets with large C, we relax the optimisation on the continuous component of the input, uniformly sampling 200 samples for each BO iteration (compared to 5000 samples required for the CoCaBO baseline). As shown in the results, 200 samples was sufficient in practice for VPBO to achieve convergence and by relaxing the optimisation on the continuous component, we are able to achieve improved mean wall-clock time over CoCaBO (refer to Appendix D). We follow the optimisation approach of Ru et al. (2020), optimising GP hyperparameters by maximising the log marginal likelihood every 10 iterations using multi-started gradient descent, including the mixed-kernel hyperparameter (*i.e.* $\lambda = \texttt{auto}$).

For our initialisation budget, we also allocate an initial 24 observation samples. Since our method relies on the surrogate for making choices on both the categorical and continuous parts of the input, it is important that the GP model is initialised correctly. In particular, a good choice of initial observation data can significantly improve the performance of our VPBO method. To ensure an appropriate choice of

initial observations is made for our surrogate GP model, we employ a search initialisation procedure, where we use the BO back-end to search for $\hat{\boldsymbol{x}}_0^*$. The search initialisation process is described as follows: we first randomly sample up to half the initial observation budget (*i.e.* 12 points) as usual, randomly selecting $\boldsymbol{h}_0^j$ and $\boldsymbol{x}_0^j$ where $j \in \{1, ..., 12\}$. We then continue to randomly select $\boldsymbol{h}_0^j$, whilst using the BO back-end to search for $\boldsymbol{x}_0^j$ for the remaining initial observation budget (*i.e.* $j \in \{13, ..24\}$).

To search for $\boldsymbol{x}_0^j$, we use the Max-Value Entropy Search (MES) acquisition metric of Wang & Jegelka (2017), which has been shown to be efficient at optimising output space information gain and fulfils the criteria we are interested in during the initial observation stage. In Sec. 5.3, we perform a detailed ablation study on this search initialisation method, comparing our method to the CoCaBO baseline.

Datasets   For all benchmarks, the continuous inputs were normalised to $\boldsymbol{x} \in [0, 1]^{d_x}$ and all experiments were conducted on an 8-core 3.4Ghz Intel Xeon processor with 64GB RAM. Our benchmarks include a variety of synthetic and real, single and multi-objective problems:

- **Func-2C** is a synthetic test problem with two continuous variables ($d_x = 2$) and two categorical variables ($k = 2$). The categorical variables determine the linear combination of three $2D$ global optimisation benchmark functions: Beale, Six-Hump Camel and Rosenbrock[2] (15 possible

---

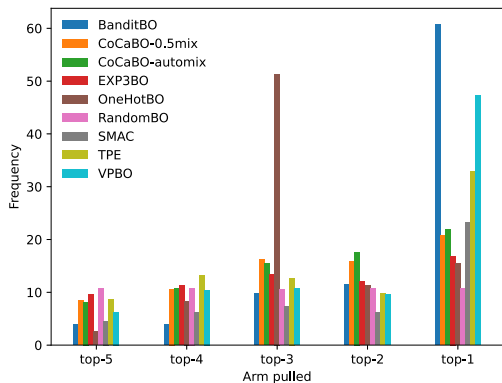[2]https://www.sfu.ca/~ssurjano/optimization.html

Figure 2. Top 5 average best arm pull frequency of all methods across all 7 datasets.

combinations).

- **Func-3C** is an extension of **Func-2C** with an additional categorical variable ($k = 3$), allowing for more possible categorical choices (60 possible combinations).

- **Reizman-Suzuki** represents virtual experiments for the Suzuki-Miyaura Cross-Coupling reaction where experimental outcomes are based on an emulator that is trained on the experimental data published by Reizman et al. (2016). The experimental emulator is provided by (Felton et al., 2020) and outputs product yield and catalyst turnover number as objectives to be maximised. This optimisation problem consists of three continuous variables ($d_x = 3$): temperature, residence time and catalyst loading; and one categorical variable ($k = 1$): catalyst choice (8 possible combinations).

- **Baumgartner** represents virtual experiments for the Aniline Cross-Coupling reaction where experimental outcomes are based on an emulator that is trained on the experimental data published by Baumgartner et al. (2019). The experimental emulator is provided by (Felton et al., 2020) and outputs product yield and material cost as objectives to be maximised and minimised respectively. This optimisation problem consists of three continuous variables ($d_x = 3$): temperature, residence time and base equivalents; and two categorical variables ($k = 2$): catalyst and base choices (12 possible combinations).

- **SVM-Boston** outputs the test negative mean square error from a support vector machine (SVM) for regression on the Boston Housing dataset (Dua et al., 2017). This optimisation problem consists of three continuous variables ($d_x = 3$): nu value, error term penalty and tolerance; and three categorical variables ($k = 3$): kernel type, kernel coefficient and shrinking (16 possible combinations).

- **XG-MNIST** outputs the test classification accuracy of an XGBoost model (Chen et al., 2015) on MNIST (Le-Cun, 1998). This optimisation problem consists of four continuous variables ($d_x = 4$): learning rate, minimum split loss, subsampling ratio and regularisation; with four categorical variables ($k = 4$): booster type, growth policy, training objective and maximum tree depth (80 possible combinations).

- **NAS-Bench-101** outputs the test classification accuracy from performing architecture search on the convolutional neural network topology for CIFAR-10 image classification (Krizhevsky et al., 2014). The search was conducted using the NAS-Bench-101 dataset using the proposed search space from Ying et al. (2019). This optimisation problem consists of 22 continuous variables ($d_x = 22$) corresponding to edge probabilities and five categorical variables ($k = 5$) of intermediate node types (243 possible combinations).

5.1. Performance of VPBO

We evaluated the optimisation performance of our proposed VPBO method against existing methods. Following Ru et al. (2020), we set each optimisation trial to consist of $T = 200$ iterations. We performed 20 random trials for the Func2C, Func3C, Reizman-Suzuki and Baumgartner datasets. For the SVM-Boston, XG-MNIST and NASBench-101 datasets, we performed 10 random trials. Means and standard errors over all trials are presented in Fig. 1.

The gray dotted line in Fig. 1 represents an Oracle agent that we trained (similar to Gopakumar et al. (2018)). Here, we run BO for each of the C choices of possible category combinations, with each choice allocated its own separate GP surrogate model GP_c. We allocate each of the C choices the full $T = 200$ iterations as well as all 24 initial sample points and optimise the hyperparameters for GP_c every iteration by maximising the log marginal likelihood. At each iteration t, the Oracle's performance is then taken as the best value for f over all possible categorical choices C. Hence, we use the Oracle to represent a best possible outcome scenario at each iteration t.

Across all synthetic and real-world problems, our VPBO method outperforms other competing approaches, where the general trend is that VPBO demonstrates a significant improvement in initial performance and more quickly converges towards the performance of the Oracle agent when compared with the competing baselines. Note that due to the nature of the EXP3BO and Bandit-BO methods maintaining separate surrogate models for each of the possible C categories and being allocated 3 initial samples per surrogate, for problems where C is large (*e.g.* Func3C, XG-MNIST and NASBench-101), their initial performance at $t = 1$ is
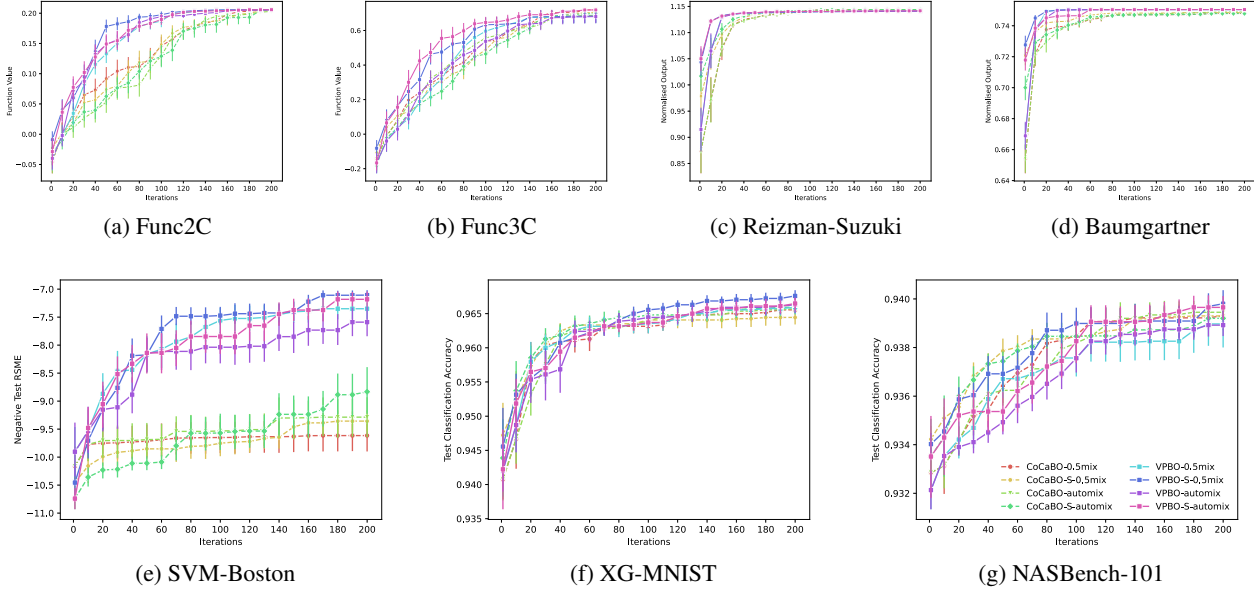
Figure 3. Ablation study on search initialisation procedure between CoCaBO and our proposed VPBO method

significantly higher due to the larger number of initial samples observed from f (*e.g.* 180 samples for Func3C vs. 24 for all other methods including VPBO).

5.2. Choice of Categorical Variables

We compare our VPBO method against existing baselines in terms of how often they make the best selection of categorical variables. Using bandit problem terminology, we term each combination of categories an arm and selecting a combination of categories which corresponds to a particular arm is referred to as "pulling" that arm. The best selection is defined as the set of choices found to yield the best results according to the Oracle agent described in Sec. 5.1. Ideally, our agent should choose the best performing arm often, so that the continuous component of the input is given many opportunities to find the optimal x_t^* at any given iteration t. Fig. 2 shows the average arm pull frequency of VPBO and all baselines on the top 5 performing arms as indicated by the Oracle, averaged across the 7 datasets. We can see that VPBO outperforms all baselines except Bandit-BO, where it offers competitive results. Note that due to the nature of the Bandit-BO algorithm, it must be allocated significantly more initial observation samples (as detailed in Sec. 5) before an optimisation trial, when compared to VPBO.

5.3. Ablation on Search Initialisation

We perform an ablation study of the effect of our search initialisation strategy when compared to the typical fully random initialisation approach. The results across the 7

datasets are shown in Fig. 3. Here, we compare our method to the baseline of Ru et al. (2020), selecting the two best performing variants, 0.5mix and automix. The former fixes $\lambda = 0.5$, while the latter learns λ as part of the GP hyperparameters. CoCaBO-S and VPBO-S indicate where search initialisation has been employed. Using search initialisation significantly improves the performance of our VPBO method but in general makes a negligible difference in CoCaBO. As shown in Fig. 3, our VPBO variants tend to outperform CoCaBO variants regardless of the initialisation procedure.

6. Conclusion

In this paper, we presented a mixed-variable black-box optimisation approach which adopts a global BO approach in the joint optimisation of categorical and continuous variables. Our solution uses a single acquisition criterion for optimising the mixed input input space under a unified framework. We analyse our method theoretically and empirically, demonstrating that VPBO has a sub-linear bound on regret and shows significant improvements in performance over existing mixed-variable optimisation approaches on a wide range of problems. Future work would be to extend our model into additional search spaces such as graphs and into higher dimensional categorical settings.

References

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2):235–256, 2002.

Baumgartner, L. M., Dennis, J. M., White, N. A., Buchwald, S. L., and Jensen, K. F. Use of a droplet platform to optimize pd-catalyzed c–n coupling reactions promoted by organic bases. Organic Process Research & Development, 23(8):1594–1601, 2019.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. Advances in neural information processing systems, 24, 2011.

Breiman, L. Random forests. Machine learning, 45(1):5–32, 2001.

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

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. Online optimization in x-armed bandits. In Twenty-Second Annual Conference on Neural Information Processing Systems, 2008.

Bull, A. D. Convergence rates of efficient global optimization algorithms. Journal of Machine Learning Research, 12(10), 2011.

Chapelle, O. and Li, L. An empirical evaluation of thompson sampling. Advances in neural information processing systems, 24:2249–2257, 2011.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., et al. Xgboost: extreme gradient boosting. R package version 0.4-2, 1(4):1–4, 2015.

Dua, D., Graff, C., et al. Uci machine learning repository. 2017.

Dudley, R. M. Sample functions of the gaussian process. Selected Works of RM Dudley, pp. 187–224, 2010.

Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., and Zoubin, G. Structure discovery in nonparametric regression through compositional kernel search. In International Conference on Machine Learning, pp. 1166–1174. PMLR, 2013.

Felton, K., Rittig, J., and Lapkin, A. Summit: Benchmarking machine learning methods for reaction optimisation. 2020.

Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q., and Cunningham, J. P. Bayesian optimization with inequality constraints. In ICML, volume 2014, pp. 937–945, 2014.

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.

Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., and Sculley, D. Google vizier: A service for black-box optimization. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1487–1495, 2017.

González, J., Dai, Z., Hennig, P., and Lawrence, N. Batch bayesian optimization via local penalization. In Artificial intelligence and statistics, pp. 648–657. PMLR, 2016.

Gopakumar, S., Gupta, S., Rana, S., Nguyen, V., and Venkatesh, S. Algorithmic assurance: An active approach to algorithmic testing using bayesian optimisation. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 5470–5478, 2018.

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

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. Springer, 2011.

Hutter, F., Kotthoff, L., and Vanschoren, J. Automated machine learning: methods, systems, challenges. Springer Nature, 2019.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4):455–492, 1998.

Kandasamy, K., Schneider, J., and Póczos, B. High dimensional bayesian optimisation and bandits via additive models. In International conference on machine learning, pp. 295–304. PMLR, 2015.

Kirchhoff, D. and Kuhnt, S. Gaussian process models with low-rank correlation matrices for both continuous and categorical inputs. arXiv preprint arXiv:2010.02574, 2020.

Krause, A. and Ong, C. S. Contextual gaussian process bandit optimization. In Nips, pp. 2447–2455, 2011.

Krizhevsky, A., Nair, V., and Hinton, G. The cifar-10 dataset. online: http://www. cs. toronto. edu/kriz/cifar. html, 55 (5), 2014.

LeCun, Y. The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/, 1998.

Mockus, J., Tiesis, V., and Zilinskas, A. The application of bayesian methods for seeking the extremum. Towards global optimization, 2(117-129):2, 1978.

Nguyen, D., Gupta, S., Rana, S., Shilton, A., and Venkatesh, S. Bayesian optimization for categorical and category-specific continuous inputs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 5256–5263, 2020.

Nguyen, V., Gupta, S., Rana, S., Li, C., and Venkatesh, S. Regret for expected improvement over the best-observed value and stopping condition. In Asian Conference on Machine Learning, pp. 279–294. PMLR, 2017.

Orr, M. J. et al. Introduction to radial basis function networks, 1996.

Rana, S., Li, C., Gupta, S., Nguyen, V., and Venkatesh, S. High dimensional bayesian optimization with elastic gaussian process. In International conference on machine learning, pp. 2883–2891. PMLR, 2017.

Rasmussen, C. E. Gaussian processes in machine learning. In Summer school on machine learning, pp. 63–71. Springer, 2003.

Reizman, B. J., Wang, Y.-M., Buchwald, S. L., and Jensen, K. F. Suzuki–miyaura cross-coupling optimization enabled by automated feedback. Reaction chemistry & engineering, 1(6):658–666, 2016.

Ru, B., Alvi, A., Nguyen, V., Osborne, M. A., and Roberts, S. Bayesian optimisation over multiple continuous and categorical inputs. In International Conference on Machine Learning, pp. 8276–8285. PMLR, 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, 2015.

Shields, B. J., Stevens, J., Li, J., Parasram, M., Damani, F., Alvarado, J. I. M., Janey, J. M., Adams, R. P., and Doyle, A. G. Bayesian reaction optimization as a tool for chemical synthesis. Nature, 590(7844):89–96, 2021.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems, 25, 2012.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. arXiv preprint arXiv:0912.3995, 2009.

Vazquez, E. and Bect, J. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. Journal of Statistical Planning and inference, 140(11):3088–3095, 2010.

Vermorel, J. and Mohri, M. Multi-armed bandit algorithms and empirical evaluation. In European conference on machine learning, pp. 437–448. Springer, 2005.

Wang, Z. and Jegelka, S. Max-value entropy search for efficient bayesian optimization. In International Conference on Machine Learning, pp. 3627–3635. PMLR, 2017.

Xu, Z., Kersting, K., and Tresp, V. Multi-relational learning with gaussian processes. In Twenty-First International Joint Conference on Artificial Intelligence, 2009.

Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., and Hutter, F. Nas-bench-101: Towards reproducible neural architecture search. In International Conference on Machine Learning, pp. 7105–7114. PMLR, 2019.

Zhou, Q., Qian, P. Z., and Zhou, S. A simple approach to emulation for computer models with qualitative and quantitative factors. Technometrics, 53(3):266–273, 2011.

Zhou, Z., Li, X., and Zare, R. N. Optimizing chemical reactions with deep reinforcement learning. ACS central science, 3(12):1337–1344, 2017.

A. Derivation of the Closed-form for Expected Improvement

Let $\mathcal{D}_T = \{x_t \in \mathbb{R}^d, y_t \in \mathbb{R}\}_{t=1}^T$ be the set of observations up to time T. The improvement function at iteration t is given by:

$$\mathcal{I}_t(\boldsymbol{x}_t) = \max\{0, f(\boldsymbol{x}_t) - \mathcal{E}_t\} \qquad (14)$$

where $\mathcal{E}_t$ defines an incumbent at iteration t, such that $\mathcal{E}_t = y_{t-1}^{max} = \max_{y_i \in \mathcal{D}_{t-1}} y_i$. For concise notation, we drop the $\boldsymbol{x}$ term and denote $\mathcal{I}$ as the improvement function, μ as the predictive mean of the posterior and σ as the predictive variance of the posterier. The likelihood of improvement $\mathcal{I}$ on a normal posterior distribution is given as:

$$Pr(\mathcal{I}) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(\mu - \mathcal{E} - \mathcal{I})^2}{2\sigma^2}\right) \qquad (15)$$

The expected improvement is defined as $\alpha^{EI}(\boldsymbol{x}) = \mathbb{E}[\mathcal{I}(\boldsymbol{x})]$. Using Eq. 15, we can rewrite the expected improvement as:

$$\alpha^{EI}(\boldsymbol{x}) = \int_0^\infty \frac{\mathcal{I}}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(\mu - \mathcal{E} - \mathcal{I})^2}{2\sigma^2}\right)d\mathcal{I} \quad (16)$$

Let $t = \frac{\mu - \mathcal{E} - \mathcal{I}}{\sigma}$, then $\mathcal{I} = -t\sigma + \mu - \mathcal{E}$ and $dt = -\frac{1}{\sigma}d\mathcal{I}$. We can rewrite $\alpha^{EI}(\boldsymbol{x})$ as:

$$\alpha^{EI}(\boldsymbol{x}) = \int_{t=\frac{\mu-\mathcal{E}}{\sigma}}^{-\infty} \frac{-t\sigma + \mu - \mathcal{E}}{\sqrt{2\pi}\sigma}\exp\left(-\frac{t^2}{2}\right)(-\sigma)dt$$

$$= \int_{t=\frac{\mu-\mathcal{E}}{\sigma}}^{-\infty} \frac{t}{\sqrt{2\pi}}\exp\left(-\frac{t^2}{2}\right)dt + (\mu - \mathcal{E})\int_{-\infty}^{t=\frac{\mu-\mathcal{E}}{\sigma}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{t^2}{2}\right)dt$$

$$(17)$$

Denoting $u = t^2 = \left(\frac{\mu - \mathcal{E} - \mathcal{I}}{\sigma}\right)^2$, $du = 2tdt$, and the first term in Eq. 17 is as follows:

$$\int_{t=\frac{\mu-\mathcal{E}}{\sigma}}^{-\infty} \frac{t}{\sqrt{2\pi}}\exp\left(-\frac{t^2}{2}\right)dt = \frac{\sigma}{\sqrt{2\pi}}\int_{u=t^2}^{-\infty}\exp\left(-\frac{u}{2}\right)\frac{du}{2}$$

$$= \frac{\sigma}{\sqrt{2\pi}}\left[-\exp\left(-\frac{1}{2}\left(\frac{\mu - \mathcal{E} - \mathcal{I}}{\sigma}\right)^2\right)\right]_{\mathcal{I}=0}^{\mathcal{I}=-\infty}$$

$$= \sigma\mathcal{N}\left(\frac{\mu-\mathcal{E}}{\sigma}\Big|0,1\right) \qquad (18)$$

The second term in Eq. 17 is computed as:

$$(\mu - \mathcal{E})\int_{-\infty}^{t=\frac{\mu-\mathcal{E}}{\sigma}} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{t^2}{2}\right)dt$$

$$= (\mu - \mathcal{E})\int_{-\infty}^0 \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{\mu - \mathcal{E} - \mathcal{I}}{\sigma}\right)^2\right)d\gamma$$

$$= (\mu - \mathcal{E})\Phi\left(\frac{\mu-\mathcal{E}}{\sigma}\right)$$

$$(19)$$

Denoting $\gamma = \frac{\mu-\mathcal{E}}{\sigma}$, we obtain the closed-form for the EI acquisition function:

$$\alpha^{EI}(\boldsymbol{x}) = \sigma(\boldsymbol{x})\phi(\gamma) + (\mu(\boldsymbol{x}) - \mathcal{E})\Phi(\gamma) \qquad (20)$$

where $\phi(\gamma) = \mathcal{N}(\gamma|0,1)$ and $\Phi(\gamma)$ are the respective standard normal PDF and CDF.

B. Mixed Kernel

Categorical Kernel For our categorical kernel, we use an indicator function based on what was proposed in (Xu et al., 2009) ($1_\mathbb{C}$), where $1_\mathbb{C}$ is an indicator function which equals 1 when its condition $\mathbb{C}$ (in this case $h = h'$) is met and 0 otherwise. The categorical kernel used is defined as:

$$\kappa_h(\boldsymbol{h}, \boldsymbol{h}') = \frac{\sigma^2}{c}\sum_{i=1}^c 1_\mathbb{C} \qquad (21)$$

where $\mathbb{C}$ is the condition $h = h'$, c is the number of categories for the categorical input and σ^2 is the kernel variance. We can derive this kernel as a special case of squared exponential kernel (Orr et al., 1996). Recall the standard squared exponential kernel with unit variance being evaluated between two scalar locations x and x':

$$\kappa(x, x') = \sigma^2\exp\left(-\frac{(x-x')^2}{2\ell^2}\right) \qquad (22)$$

The lengthscale ℓ in Eq. 22 defines the similarity between two input such that as ℓ becomes smaller, the distance between locations considered similar (*i.e.* high covariance) shrinks. In the limiting case of $\ell \to 0$, if two inputs are not exactly the same as each other, then no information would be provided for inferring the GP posterior's value at each other's locations. In other words, the kernel turns into the indicator function shown in Eq. 21:

$$\kappa(x, x') = \begin{cases} 1, & \text{if } x = x' \\ 0, & \text{otherwise} \end{cases} \qquad (23)$$

Thus, for each categorical variable in $\boldsymbol{h}$, we can add a squared exponential kernel with $\ell \to 0$.

Combining kernels For combining kernels, we can consider several techniques which result in valid kernels (Duvenaud et al., 2013). The most straight-forward approach of summing the categorical and continuous kernels together has been used previously in high dimensional BO and bandits using additive models (Kandasamy et al., 2015). However, adding the continuous kernel to the categorical kernel (*i.e.* $\kappa_h(\boldsymbol{h}, \boldsymbol{h}') + \kappa_x(\boldsymbol{x}, \boldsymbol{x}')$) provided limited expressiveness in the resulting kernel. In practice, this equates to learning a single common trend over $\boldsymbol{x}$ with an offset depending on $\boldsymbol{h}$.

Alternatively, we can consider using the product of kernels (*i.e.* $\kappa_h(\boldsymbol{h}, \boldsymbol{h}') \times \kappa_x(\boldsymbol{x}, \boldsymbol{x}')$) to improve the expressiveness of the resulting kernel. Multiplicatively combining kernels in this manner allows the encoding of couplings between continuous and categorical domains, which enables a richer set of relationships to be captured. However, during the earlier iterations of BO where there are unlikely to be overlapping categories in the data, the resulting product kernel will be zero which would prevent the model from learning.

Ru et al. (2020) proposed a mixed-kernel as part of their CoCaBO framework which is a mixture of sum and product kernels:

$$\kappa_z(\boldsymbol{z}, \boldsymbol{z}') = (1 - \lambda)(\kappa_h(\boldsymbol{h}, \boldsymbol{h}') + \kappa_x(\boldsymbol{x}, \boldsymbol{x}'))$$
$$+ \lambda \kappa_h(\boldsymbol{h}, \boldsymbol{h}')\kappa_x(\boldsymbol{x}, \boldsymbol{x}') \tag{24}$$

where a trade-off between the two kernels is controlled by a hyperparameter $\lambda \in [0, 1]$ and can be jointly optimised with the GP hyperparameters. For our VPBO method, we adopt the same mixed-kernel as in Ru et al. (2020).

C. Consistency in Performance

Additionally, we inspect the rate at which our VPBO method makes a "good" choice of categorical variables. Here, we define a good choice as selecting a categorical combination which results in a returned reward which falls within a range of 95% of the "best" reward as indicated by the Oracle agent. Fig. 4 shows the average frequency along with standard error of making a good choice categorical variables over optimisation iterations across 20 optimisation trials for synthetic datasets and 10 optimisation trials for real datasets.

Here, we are interested in seeing if each method is able to consistently make good choices for selecting categorical variables which contain high potential for obtaining large rewards. We can see that our VPBO is able to consistently select categorical choices with high potential and obtain higher rewards on average compared to the other baselines. Bandit-BO is able to outperform VPBO across some of the datasets, although we note that Bandit-BO is given a much larger set of initial observations than VPBO. Similarly, CoCaBO is able to outperform our VPBO method on the synthetic datasets, but CoCaBO does not perform very consistently across the multi-objective and real-world datasets.

D. Wall-clock Time Overhead

We measure the mean wall-clock time overhead of VPBO compared to several baseline methods, averaged across 200 optimisation iterations. Table 1 shows the average wall-clock time overhead for an optimisation iteration across all 7 datasets for several methods including VPBO. For each method, the wall-clock time overhead was bench-marked using the same machine. We can see that our VPBO offers comparable wall-clock time overhead with competing baselines and in particular, improved wall-clock time overhead in comparison to the CoCaBO baseline which it is most similar to from a model perspective. Differences in such overhead (a few seconds) is negligible when compared to function evaluation time where the scale of time taken is often in hours or even days.
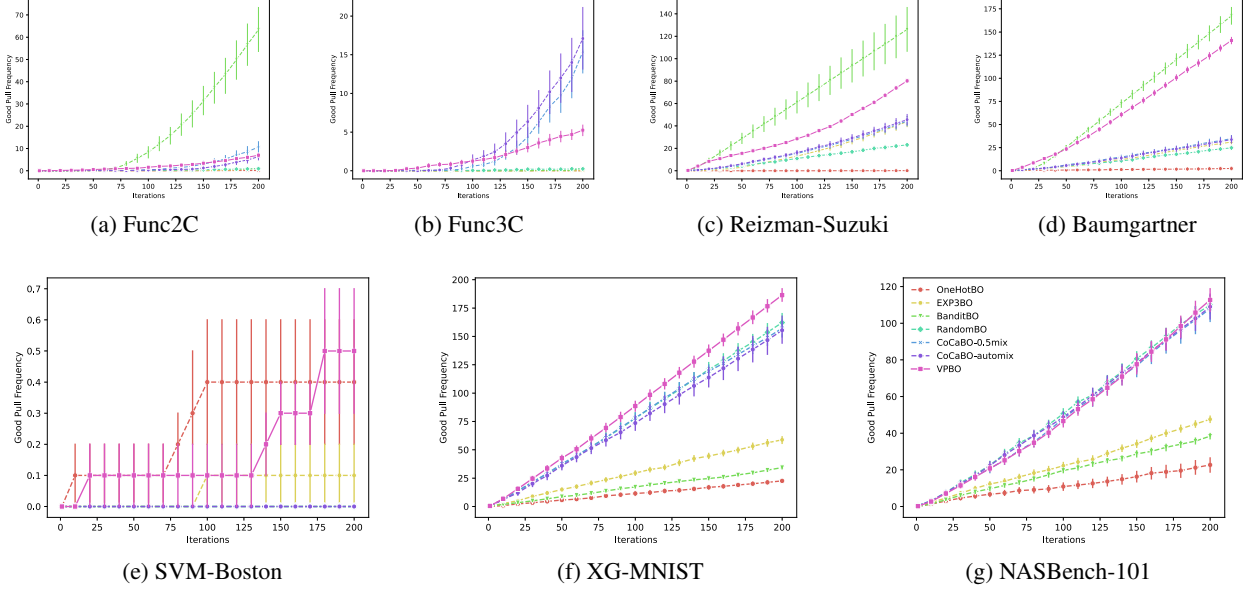
Figure 4. Average frequency of making a good choice of categorical variables for several methods on synthetic and real-world tasks.

	Method					
Dataset	One Hot BO	RandomBO	EXP3BO	BanditBO	CoCaBO	VPBO
Func2C	0.7492	0.7525	0.3020	0.4206	0.9600	0.5579
Func3C	0.8392	0.7629	0.2553	0.3388	0.9118	1.0018
Reizman-Suzuki	0.6861	1.0428	0.4380	0.5951	1.1590	0.5144
Baumgartner	1.1952	0.8412	0.3575	0.5056	1.0762	0.5627
SVM-Boston	0.6964	0.8584	0.3550	0.3068	1.0042	0.5504
XG-MNIST	1.0405	3.0324	2.4074	2.5534	3.5881	5.0091
NASBench-101	0.8771	2.2369	0.2799	10.4477	3.2122	3.0671

Table 1. The mean wall-clock time overheads for different methods for each BO iteration across 200 optimisation rounds in seconds.