Bayesian Optimization over Discrete and Mixed Spaces via Probabilistic Reparameterization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Optimizing expensive-to-evaluate black-box functions of discrete (and potentially continuous) design parameters is a ubiquitous problem in scientific and engineering applications. Bayesian optimization (BO) is a popular sample-efficient method that selects promising designs to evaluate by optimizing an acquisition function (AF) over some domain based on a probabilistic surrogate model. However, maximizing the AF over mixed or high-cardinality discrete search spaces is challenging as we cannot use standard gradient-based methods or evaluate the AF at every point in the search space. To address this issue, we propose using probabilistic reparameterization (PR). Instead of directly optimizing the AF over the search space containing discrete parameters, we instead maximize the expectation of the AF over a probability distribution defined by continuous parameters. We prove that under suitable probability distributions, the BO policy that maximizes the probabilistic objective is the same as that which maximizes the AF, and therefore, PR enjoys the same regret bounds as the underlying AF. Moreover, our approach admits *provably convergent global optimization of the AF* (an often neglected requisite for commonly-used BO regret bounds) using scalable, unbiased estimators of both the probabilistic objective and its gradient. We validate our approach empirically and demonstrate state-of-the-art optimization performance on many real-world applications. PR is complementary to (and benefits) recent work and naturally generalizes to settings with multiple objectives and black-box constraints.

1 Introduction

Many scientific and engineering problems involve tuning discrete and/or continuous parameters to optimize an objective function. Often, the objective function is "black-box", meaning it has no known closed-form expression. For example, optimizing the design of an electrospun oil sorbent—a material that can be used to absorb oil in the case of a marine oil spill to mitigate ecological harm—can involve tuning both discrete ordinal experimental conditions and continuous parameters controlling the composition of the material to maximize the oil absorption capacity, mechanical strength, and water contact angle [47]. For another example, optimizing the structural design of a welded beam can involve tuning categorical parameters such as the type of metal, binary parameters such as the welding type, and discrete ordinal parameters such as the dimensions of the different components of the beam–resulting in a search space with over 370 million possible designs [43]. We consider the scenario where querying the objective function is expensive, in which case sample-efficiency is crucial. In the case of designing the oil sorbent, evaluating the objective function requires manufacturing the material and measuring its properties in a laboratory, requiring significant time and resources.

Bayesian optimization (BO) is a popular technique for sample-efficient black-box optimization, due to its proven performance guarantees in many settings [41] and its strong empirical performance [18]. BO leverages a probabilistic surrogate model of the unknown objective(s) and an acquisition function

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

(AF) that provides utility values for evaluating a new design to balance exploration and exploitation. Typically, the maximizer of the AF is selected as the next design to evaluate. However, maximizing the AF over mixed search spaces (i.e., those consisting of discrete and continuous parameters) or large discrete search spaces is challenging[1] because continuous (or gradient-based) optimization routines cannot be directly applied. Theoretical performance guarantees of BO policies require that the maximizer of the acquisition function is found and selected as the next design to evaluate on the black-box objective function [41]. When the maximizer is not found, regret properties are not guaranteed, and the performance of the BO policy may degrade.

To tackle these challenges, we propose a technique for improving AF optimization using a probabilistic reparameterization (PR) of the discrete parameters. Our main contributions are:

1. We use PR to optimize a probabilistic objective (PO): the expectation of the AF over an induced distribution over the discrete parameters.
2. We prove that the maximizers of the acquisition function are a surjective function of the maximizers of the PO and hence, the policy that chooses designs that are best with respect to the PO enjoys the same performance guarantees as the standard BO policy.
3. We derive scalable, unbiased Monte Carlo estimators of the probabilistic objective and its gradient with respect to the parameters of the induced distribution. We show that stochastic gradient ascent using our gradient estimator is guaranteed to converge to a stationary point on the PO surface and will recover a global maximum of the underlying AF as the number of starting points increases. This is important because finding the global maximizer of the AF is fundamental for common BO regret bounds [41]. Empirically, we find that with a modest number of starting points, PR leads to better AF optimization than alternative approaches.
4. We show that PR yields state-of-the-art optimization performance on a wide variety of real-world design problems with discrete and mixed search spaces. Importantly, PR is *complementary* to many existing approaches such as popular multi-objective, constrained, and trust region-based approaches; in particular, PR is agnostic to the underlying probabilistic model over discrete parameters—which is not the case for many alternative methods.

2 Preliminaries

Bayesian Optimization We consider the problem of optimizing a black-box function $f : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ over a compact search space $\mathcal{X} \times \mathcal{Z}$, where $\mathcal{X} = \mathcal{X}^{(1)} \times \cdots \times \mathcal{X}^{(d)}$ is the domain of the $d \geq 0$ continuous parameters ($x^{(i)} \in \mathcal{X}^{(i)}$ for $i = 1, ..., d$) and $\mathcal{Z} = \mathcal{Z}^{(1)} \times \cdots \times \mathcal{Z}^{(d_z)}$ is the domain of the $d_z \geq 1$ discrete parameters ($z^{(i)} \in \mathcal{Z}^{(i)}$ for $i = 1, ..., d_z$).[2]

When the black-box function is expensive-to-evaluate, Bayesian optimization (BO) is popular owing to its sample efficiency. BO leverages (i) a probabilistic surrogate model—typically a Gaussian process (GP) [35]—fit to a data set $\mathcal{D}_n = \{\boldsymbol{x}_i, \boldsymbol{z}_i, y_i\}_{i=1}^n$ of designs and corresponding (potentially noisy) observations $y_i = f(\boldsymbol{x}_i, \boldsymbol{z}_i) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)$, and (ii) an acquisition function $\alpha(\boldsymbol{x}, \boldsymbol{z})$ that uses the surrogate model's posterior distribution to quantify the value of evaluating a new design. Common AFs include expected improvement (EI) [25] and upper confidence bound (UCB) [41]—the latter of which enjoys no-regret guarantees in certain settings. The next design to evaluate is chosen by maximizing the AF $\alpha(\boldsymbol{x}, \boldsymbol{z})$ over $\mathcal{X} \times \mathcal{Z}$. Although the black-box objective f is expensive-to-evaluate, the AF is relatively cheap-to-query, and therefore, can be optimized numerically (gradient-based optimization routines are often used to maximize the AF over continuous domains).

Discrete Parameters Typically, BO assumes that the inputs are continuous. However, discrete parameters such as binary parameters, discrete ordinal parameters, and non-ordered categorical parameters, which are summarized in Table 1, are ubiquitous in many applications. In the presence of such parameters, optimizing the AF is more difficult, as standard gradient-based approaches cannot be directly applied. Recent works have proposed various approaches including multi-armed bandits

[1] If the discrete search space has low enough cardinality that the AF can be evaluated at every discrete element, then acquisition optimization can be solved trivially.

[2] Throughout this paper, we use a mixed search space $\mathcal{X} \times \mathcal{Z}$ in our derivations, theorems, and proofs, without loss of generality with respect to the case of a purely discrete search space. If $d = 0$, then the objective function $f : \mathcal{Z} \to \mathbb{R}$ is defined over the discrete space $\mathcal{Z}$ and the continuous parameters in this exposition can simply be ignored.

[31, 37] and local search [32] for discrete domains and interleaved discrete/continuous optimization procedures for mixed domains [12, 45]. A simple and widely-used approach across many popular BO packages [1, 42] is to one-hot encode the categorical parameters, apply a continuous relaxation when solving the optimization, and discretize (round) the resulting continuous candidates. Examples of continuous relaxations and discretization functions are listed in Table 1.

Table 1: Different parameter types, their continuous relaxations, and discretization functions.

TYPE	DOMAIN	CONT. RELAXATION	discretize$(\cdot)$ FUNCTION
BINARY	$z \in \{0, 1\}$	$z' \in [0, 1]$	$\text{round}(z')$
ORDINAL	$z \in \{0, ..., C-1\}$	$z' \in [0, C-1]$	$\text{round}(z')$
CATEGORICAL	$z \in \{0, ..., C-1\}$	$z' \in [0, 1]^C$	$\arg\max_c z'^{(c)}$

Although using a continuous relaxation allows for efficient optimization using standard optimization routines in an alternate continuous domain $\mathcal{Z}' \subset \mathbb{R}^m$, the acquisition value for an infeasible continuous value (i.e., $z' \notin \mathcal{Z}$) does not account for the discretization that must occur before the black-box function is evaluated. Moreover, the acquisition value for an infeasible continuous value can be larger than the acquisition value after discretization. For an illustration of this, see Fig. 1 (middle/right). In the worst case, BO will repeatedly select the same infeasible continuous design due to its high acquisition value, but discretization will result in a design that has already been evaluated and has zero acquisition value. To mitigate this degenerate behavior, Garrido-Merchán and Hernández-Lobato [20] propose to discretize the candidate before evaluating the AF. While this improves performance on small search spaces, the resulting AF and surrogate have large flat regions, which makes it difficult to optimize the AF. The authors of [20] propose to approximate the gradients using finite differences, but, empirically, we find that this approach to be suboptimal for optimizing the AF.

3 Probabilistic Reparameterization

We propose an alternative approach based on probabilistic reparameterization, a relaxation of the original optimization problem involving discrete parameters. Rather than directly optimizing the AF via a continuous relaxation $\boldsymbol{z}'$ of the design $\boldsymbol{z}$, we instead reparameterize the optimization problem by introducing a discrete probability distribution $p(\boldsymbol{Z}|\boldsymbol{\theta})$ over a random variable $\boldsymbol{Z}$ with support exclusively over $\mathcal{Z}$. This distribution is parameterized by a vector of continuous parameters $\boldsymbol{\theta}$.

Table 2: Examples of probabilistic reparameterizations for different parameter types. Although the distribution $p(Z|\theta)$ for an ordinal random variable Z does not come from a standard probability distribution, the random variable B is Bernoulli and for any θ, the distribution $p(Z|\theta)$ is simply a Bernoulli random variable with an additive offset.

PARAMETER TYPE	RANDOM VARIABLE	CONTINUOUS PARAMETER
BINARY	$Z \sim \text{BERNOULLI}(\theta)$	$\theta \in [0, 1]$
ORDINAL	$Z = \lfloor \theta \rfloor + B, B \sim \text{BERNOULLI}(\theta - \lfloor \theta \rfloor)$	$\theta \in [0, C-1]$
CATEGORICAL	$Z \sim \text{CATEGORICAL}(\theta), \theta = (\theta^{(1)}, ..., \theta^{(C)})$	$\theta \in [0, 1]^C$

We use $\boldsymbol{z}$ to denote the vector $(z^{(1)}, ..., z^{(d_z)})$, where each element is a different (possibly vector-valued discrete parameter). Given this reparameterization, the probabilistic objective (PO) is defined as:

$$\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]. \tag{1}$$

PR enables optimizing $\boldsymbol{\theta}$ and $\boldsymbol{x}$ over a continuous space to maximize the PO instead of optimizing $\boldsymbol{x}, \boldsymbol{z}$ to maximize α directly over the mixed search space $\mathcal{X} \times \mathcal{Z}$. As we will show later, the maxima of the PO (when discretized via Table 1) recovers the true optimizers of the AF that would be obtained if the search over $\mathcal{X} \times \mathcal{Z}$ were to converge globally. Choosing $p(\boldsymbol{Z}|\theta)$ to be a discrete distribution over $\mathcal{Z}$ means the realizations of $\boldsymbol{Z}$ are feasible values in $\mathcal{Z}$. Hence, the AF is only evaluated for feasible discrete designs. Since $p(\boldsymbol{Z}|\theta)$ is a discrete probability distribution, we can express $\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\theta)}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$ as a linear combination where each discrete design is weighted by its probability mass:

$$\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\theta)}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{\boldsymbol{z} \in \mathcal{Z}} p(\boldsymbol{Z}|\theta)\alpha(\boldsymbol{x}, \boldsymbol{Z}), \tag{2}$$

Example distributions for binary, ordinal, and categorical parameters are provided in Table 2. Although ordinal parameters could use the same categorical distributions as the non-ordered categorical parameters, we opt for the provided proposal distribution since it uses a scalar θ, rather than a C-element vector. Using an independent random variable $Z^{(i)} \sim p(Z^{(i)}|\theta^{(i)})$ for each parameter $z^{(i)}$ for $i = 1, ..., d_z$ means that the probabilistic objective can be expressed as

$$\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{z^{(1)}\in\mathcal{Z}^{(1)}} \cdots \sum_{z^{(d_z)}\in\mathcal{Z}^{(d_z)}} \alpha\big(\boldsymbol{x}, z^{(1)}, ..., z^{(d_z)}\big) \prod_{i=1}^{d_z} p(z^{(i)}|\theta^{(i)}). \tag{3}$$

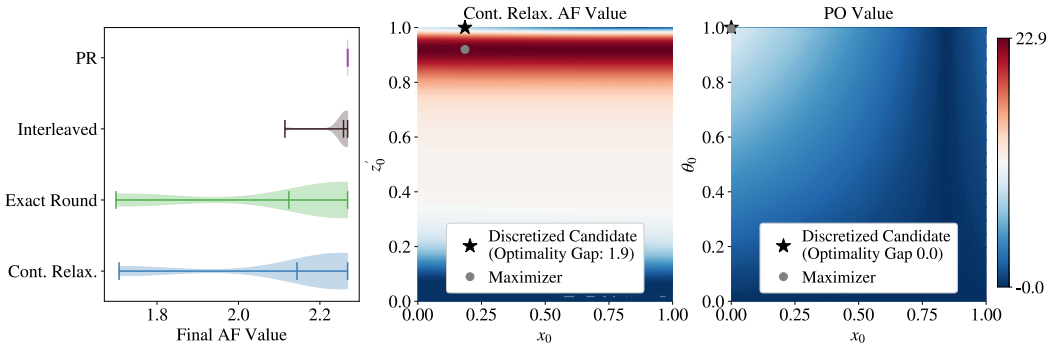

Figure 1: (**Left**) A comparison of AF optimization using different methods over a mixed search space shows that PR *outperforms alternative methods for AF optimization and has much lower variance across replications*. For each method, the best candidate across 20 restarts is selected (after discretization) and the acquisition value of the resulting feasible candidate is recorded. (**Middle/Right**) AF values with a continuous relaxation (middle) and the PO (right) for the Branin function over a mixed domain with one continuous parameter (x_0) and one binary parameter (z_0) (see Appendix C for details on Branin). (**Middle**) Under a continuous relaxation, the maximizer of the AF is an infeasible point in the domain (grey circle), which results in suboptimal AF value when rounded (black star); the optimality gap with respect to the true AF maximizer is 1.9. The maximal AF across the feasible search space is shown in white, so red regions indicated that the continuous relaxation overestimates the AF value. (**Right**) The PO is maximized at the AF unique maximizer within the valid search domain. These contours show that PR avoids the overestimation issue with a continuous relaxation.

3.1 Analytic Gradients

One important benefit of PR is that the PO in (1) is differentiable with respect to $\boldsymbol{\theta}$ (and $\boldsymbol{x}$, if the gradient of α with respect to $\boldsymbol{x}$ exists), whereas $\alpha(\boldsymbol{x}, \boldsymbol{z})$ is not differentiable with respect to $\boldsymbol{z}$. The gradients of the PO with respect to $\boldsymbol{\theta}$ and $\boldsymbol{x}$ can be obtained by differentiating Equation 2:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{\boldsymbol{z}\in\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{Z})\nabla_{\boldsymbol{\theta}} p(\boldsymbol{Z}|\boldsymbol{\theta}) \tag{4}$$

$$\nabla_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{\boldsymbol{z}\in\mathcal{Z}} p(\boldsymbol{Z}|\boldsymbol{\theta})\nabla_{\boldsymbol{x}} \alpha(\boldsymbol{x}, \boldsymbol{Z}) \tag{5}$$

This enables optimizing the PO (line 4 of Algorithm 1) efficiently and effectively using gradient-based methods over discrete or mixed search spaces.

3.2 Theoretical Properties

In this section, we derive theoretical properties of PR. We consider the case of a single discrete parameter for simplicity, but the theorems and proofs are easily extended to handle multiple discrete parameters as well as continuous parameters. Proofs are provided in Appendix B. Our first result is that the maximizers of the PO, when discretized according to the discretization functions in Table 1, are the maximizers of the AF over $\mathcal{X} \times \mathcal{Z}$.[3]

[3]Discretization is only necessary to address the situation where there is not a single unique maximizer of the AF. In such a scenario, there can be infinitely many optimal values of $\boldsymbol{\theta}$. In order to obtain an optimal $\boldsymbol{z}$, we discretize an optimal $\boldsymbol{\theta}$ deterministically.

Theorem 1 (Consistent Maximizers)**.** *Let $\mathcal{H}^*$ be the maximizers of $\alpha(\boldsymbol{x}, \boldsymbol{z})$: $\mathcal{H}^* = \{(\boldsymbol{x}, \boldsymbol{z}) \in \arg\max_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{X}\times\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})\}$. Let $\mathcal{J}^* \subseteq \mathcal{X} \times \Theta$ be the maximizers of $\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$: $\mathcal{J}^* = \{(\boldsymbol{x}, \boldsymbol{\theta}) \in \arg\max_{(\boldsymbol{x},\boldsymbol{\theta})\in\mathcal{X}\times\Theta} \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]\}$, where Θ is the domain of $\boldsymbol{\theta}$. Let $\hat{\mathcal{H}}^* \subseteq \mathcal{X} \times \mathcal{Z}$ be the maximizers of $\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$ where $\boldsymbol{\theta}$ is discretized (component-wise) using the discretization functions in Table 1: $\hat{\mathcal{H}}^* = \{(\boldsymbol{x}, \text{discretize}(\boldsymbol{\theta})) : (\boldsymbol{x}, \boldsymbol{\theta}) \in \mathcal{J}^*\}$. Then, $\hat{\mathcal{H}}^* = \mathcal{H}^*$.*

Algorithm 1 outlines BO with probabilistic reparameterization. Importantly, we note that since $\alpha(\cdot)$ and the probabilistic reparameterization have the same set of maximizers, Algorithm 1 enjoys the performance guarantees of $\alpha(\cdot)$.

Corollary 1 (Regret Bounds)**.** *Let $\alpha(\boldsymbol{x}, \boldsymbol{z})$ be an acquisition function with bounded regret over a search space $\mathcal{X} \times \mathcal{Z}$. If the conditions for the regret bounds of $\alpha(\boldsymbol{x}, \boldsymbol{z})$ are satisfied, then Algorithm 1 with α enjoys the same regret bound.*

Corollary 1 shows that when PR is combined with an AF with theoretical performance guarantees, BO with probabilistic reparameterization enjoys the same performance guarantees. Examples of such AFs are upper confidence bound (UCB) [41] or Thompson sampling (TS) [38] for single objective optimization, and UCB or TS with Chebyshev [34] or hypervolume [53] scalarizations in the multi-objective setting.

Algorithm 1 BO with PR

Input: black-box objective $f : \mathcal{X} \times \boldsymbol{Z} \to \mathbb{R}$
Initialize $\mathcal{D}_0 \leftarrow \emptyset$, $\text{GP}_0 \leftarrow \text{GP}(\boldsymbol{0}, k)$
for $n = 1$ **to** N **do**
  $(\boldsymbol{x}_n, \boldsymbol{\theta}_n) \leftarrow \arg\max_{(\boldsymbol{x},\boldsymbol{\theta})\in\mathcal{X}\times\Theta} \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$
  $\boldsymbol{z}_n \leftarrow \text{discretize}(\boldsymbol{\theta}_n)$
  Evaluate $f(\boldsymbol{x}_n, \boldsymbol{z}_n)$
  $\mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \{(\boldsymbol{x}_n, \boldsymbol{z}_n, \boldsymbol{f}(\boldsymbol{x}_n, \boldsymbol{z}_n))\}$
  Update posterior GP_n given $\mathcal{D}_n$
end for

Although optimizing α directly should lead to the same candidate $(\boldsymbol{x}_n, \boldsymbol{z}_n)$ obtained by optimizing the PO, maximizing the AF over mixed or high-dimensional discrete search spaces is challenging because commonly used gradient-based methods cannot directly be applied. The key advantage of our approach is that maximizers of the AF can be identified efficiently and effectively by optimizing the PO using gradient information instead of directly optimizing the AF. We find that optimizing PR yields significantly better results than directly optimizing α as shown Figure 1(Left), where we compare AF optimization methods on the chemical synthesis problem [40] (see Appendix C for details). The violin plots show the distribution of final AF values and the mean. "Cont. Relax." denotes optimizing a continuous relaxation of the categoricals with exact gradients. "Exact Round" refers to optimizing a continuous relaxation with approximate gradients (via finite difference), but discretizes the relaxation before evaluating the surrogate [20]. "Interleaved" alternates between one step of local search on the discrete parameters and one step of gradient ascent on the continuous parameters (used in CASMOPOLITAN [45]). All methods except interleaved use L-BFGS-B and the expected improvement AF [25].

4 Practical Monte Carlo Estimators

4.1 Unbiased estimators of the Probabilistic Reparameterization and its Gradient

As the number of discrete configurations ($|\mathcal{Z}|$) increases, the PO and its gradient may become computationally expensive to evaluate analytically because both require a summation of $|\mathcal{Z}|$ terms. Therefore, we propose to estimate the PO and its gradient using Monte Carlo (MC) sampling. The MC estimator of the PO is given by

$$\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] \approx \frac{1}{N} \sum_{i=1}^{N} \alpha(\boldsymbol{x}, \tilde{\boldsymbol{z}}_i), \tag{6}$$

where $\tilde{\boldsymbol{z}}_1, ..., \tilde{\boldsymbol{z}}_N$ are samples from $p(\boldsymbol{Z}|\boldsymbol{\theta})$. This estimator is unbiased and can be computed for a large number of samples by evaluating the AF independently (or in chunks) for each input $(\boldsymbol{x}, \tilde{\boldsymbol{z}}_n)$.

MC can also be used to estimate the gradient of the PO with respect to $\boldsymbol{\theta}$. We opt for using a score function gradient estimator [26] (also known as REINFORCE [49]) and the likelihood ratio estimator [21]) because it is simple, scalable, and can be computed simply using the acquisition values $\{\alpha(\boldsymbol{x}, \tilde{\boldsymbol{z}}_i)\}_{i=1}^N$ that are already required for the MC estimator of the PO. Many alternative lower

variance estimators (e.g. Yin et al. [51], Yin et al. [52]) would require many additional AF evaluations (see Mohamed et al. [30] for a review of MC gradient estimation). The score function is the gradient of the log probability with respect to the parameters of the distribution: $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{Z}|\boldsymbol{\theta}) = \frac{\nabla_{\boldsymbol{\theta}} p(\boldsymbol{Z}|\boldsymbol{\theta})}{p(\boldsymbol{Z}|\boldsymbol{\theta})}$. Using this score function, we can express the analytic gradient as

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{\boldsymbol{z}\in\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{z}|\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z}) \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{Z}|\boldsymbol{\theta})].$$

The unbiased MC estimator of the gradient of the PO with respect to $\boldsymbol{\theta}$ is given by

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] \approx \frac{1}{N} \sum_{i=1}^{N} \alpha(\boldsymbol{x}, \tilde{\boldsymbol{z}}_i) \nabla_{\boldsymbol{\theta}} \log p(\tilde{\boldsymbol{z}}_i|\boldsymbol{\theta}). \tag{7}$$

Since the score function gradient is only defined when $p(\boldsymbol{z}|\boldsymbol{\theta}) > 0$, we reparameterize $\boldsymbol{\theta}$ to ensure $p(\boldsymbol{z}|\boldsymbol{\theta}) > 0$ for all $\boldsymbol{z}$ and $\boldsymbol{\theta}$ by using the softmax transformations provided in Table 3, which are commonly used for computational convenience and stability in probablistic reparameterization [51, 52]. Although θ can only be 0 or 1 when $\tau \to 0$, for any $\epsilon > 0$, there exists some τ such that the approximation error is less than ϵ. Moreover, even though $p(\boldsymbol{z}|\boldsymbol{\theta}) > 0$, when $p(\boldsymbol{z}|\boldsymbol{\theta})$ is small, a small number N of MC samples are unlikely to produce any samples where $\tilde{\boldsymbol{z}} = \boldsymbol{z}$. Instead of optimizing $\boldsymbol{\theta}$ directly, we instead optimize $\boldsymbol{\phi}$. Since the transformations $g(\cdot)$ are differentiable with respect to $\boldsymbol{\phi}$, the gradient (and MC gradient estimator) of the PO with respect to $\boldsymbol{\phi}$ are easily obtained using the gradient of the PO with respect to $\boldsymbol{\theta}$ and a simple application the chain rule (multiplying by $\nabla_{\boldsymbol{\phi}}\boldsymbol{\theta}$).

PARAMETER TYPE	TRANSFORMATION ($\theta = g(\phi)$)
BINARY	$\theta = \sigma((\phi - \frac{1}{2})/\tau)$
ORDINAL	$\theta = \lfloor\phi\rfloor + \sigma((\phi - \lfloor\phi\rfloor - \frac{1}{2})/\tau)$
CATEGORICAL	$\theta^{(c)} = \text{SOFTMAX}((\boldsymbol{\phi} - 0.5)/\tau)^{(c)}$

Table 3: Transformations where $\tau \in \mathbb{R}_+$ and $\phi, \theta \in \Theta$.

4.2 Convergence Guarantee using Stochastic Gradient Ascent

Since the score function gradient estimator is unbiased, we can leverage previous work on convergence in probability under stochastic gradient ascent [36] to arrive at our main convergence result for acquisition optimization.

Theorem 2 (Convergence Guarantee)**.** *Let $(\hat{\boldsymbol{x}}_{t,m}, \hat{\boldsymbol{\theta}}_{t,m})$ be the best solution after running stochastic gradient ascent for t time steps on the probabilistic objective $\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$ from m starting points with its unbiased MC estimators proposed above. Let $\{a_t\}_{t=1}^{\infty}$ be a sequence of positive step sizes such that $0 < \sum_t^{\infty} a_t^2 = A < \infty$, where a_t is the step size used in stochastic gradient ascent at time step t. Let $\hat{\boldsymbol{z}}_{t,m} := \text{discretize}(\hat{\boldsymbol{\theta}}_{t,m})$. Then as $t \to \infty$ and $m \to \infty$, $(\hat{\boldsymbol{x}}_{t,m}, \hat{\boldsymbol{z}}_{t,m}) \to (\boldsymbol{x}^*, \boldsymbol{z}^*) \in \arg\max_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{X}\times\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})$ with probability 1.*

The significance of Theorem 2 is that optimizing the PO is guaranteed to converge in probability to a global maximizer of the AF, meaning that optimizing the PO guarantees that resulting candidate design has maximal AF value. The implication is that the intended BO policy is followed and the underlying regret bounds of the AF are recovered (provided that the other conditions of the regret bound are met). Although global convergence is only guaranteed as $m \to \infty$, we observe in Figure 1(left) that PR yields stable, convergent acquisition optimization with only $m = 20$ starting points and outperforms alternative optimization approaches.

4.3 Deterministic Optimization via Sample Average Approximation

Although multi-start stochastic ascent is provably convergent, we opt instead to use lower variance deterministic MC estimators using common random numbers. In order maintain valid samples $\tilde{\boldsymbol{z}}_i \sim p(\boldsymbol{Z}|\boldsymbol{\theta})$ for $i = 1, ..., N$ as $\boldsymbol{\theta}$ changes over the course of the acquisition optimization, we reparameterize $\boldsymbol{Z}$ as a deterministic function $h(\cdot, \cdot)$ that operates component-wise on $\boldsymbol{\theta}$ and the random variable $\boldsymbol{U} = (u^{(1)}, ..., u^{(d_z)}), u^{(i)} \sim \text{Uniform}(0, 1)$: $\boldsymbol{Z} = h(\boldsymbol{\theta}, \boldsymbol{U})$. Descriptions of $h(\cdot, \cdot)$ for provided for each parameter type in Appendix C.2. Using a fixed a set of base samples $\{\tilde{\boldsymbol{u}}_i\}_{i=1}^{N}$, the sample average approximation estimators are obtained by substituting $\boldsymbol{z}_i = h(\boldsymbol{\theta}, \tilde{\boldsymbol{u}}_i)$ into (6) and (7). Although biased, sample average approximation estimators are deterministic, enabling the use of second-order numerical methods for efficient acquisition optimization [1]. We reduce the variance further by leveraging quasi-MC sampling [33] instead of *iid* sampling.

5 Related Work

Many methods for BO over discrete and mixed search spaces have been proposed. Previous work has largely focused on improving BO by either (i) improving the surrogate models or (ii) improving techniques for optimizing AFs.

Improving models: Historically, methods leveraging tree-based surrogate models, e.g., SMAC [23] and TPE [4], have been popular for optimizing discrete or mixed search spaces. BOCS encodes categorical parameters as binary variables and uses Bayesian linear regression with pairwise interactions [2]. COMBO uses a diffusion kernel on the graph built from a graph Cartesian product of discrete parameters [32]. MERCBO similarly exploits the combinatorial graph, but with Mercer features and Thompson sampling [13]. HYBO extends the diffusion kernels to mixed continuous-discrete spaces [12]. However, these methods scale poorly with respect to the number of data points and parameters. Moreover, of the methods listed above, only HYBO supports continuous parameters without discretization. HYBO enjoys a universal approximation property, but relies on summing over all possible orders of interactions between base kernels for each parameter which results in exponential complexity with respect to the number of parameters and limits its applicability to low-dimensional problems. GRYFFIN [22] uses kernel density estimation, but is limited to categorical search spaces. MIVABO uses a linear combination of basis functions (e.g. pseudo Boolean features [6] for discrete parameters) with interaction terms[11]. MVRSM [5] uses ReLU-based surrogates for computational efficiency, but is limited by the expressiveness of these models.

Optimizing acquisition functions: As discussed previously, Garrido-Merchán and Hernández-Lobato [20] propose using continuous relaxation and discretize the inputs before evaluating the AF. However, the resulting AF after discretization is piece-wise-flat along slices of the discrete parameters and is difficult to optimize. COCABO [37] sample discrete parameters using a multi-armed bandit and optimize the continuous parameters conditioned on the sampled discrete parameters. However, COCABO's performance degrades as number of discrete configurations increases. CASMOPOLITAN [45] uses local trust regions combined with an interleaved AF optimization strategy that alternates between local search steps on the discrete parameters and gradient ascent for the continuous parameters. Furthermore, both COCABO and CASMOPOLITAN do not inherently exploit ordinal structure. Moreover, the computational issues of many existing approaches make it difficult to apply them to multi-objective and constrained optimization.

Probabilistic reparameterization: PR has been considered for optimizing discrete parameters in other domains such as reinforcement learning [49] and sparse regression [52]. However, PR has not been leveraged for BO (although a different reparameterization is commonly used in MC AFs [50]).

Alternative methods for propagating gradients: Alternative methods for propagating gradients through discrete structures have been considered in the deep learning community (among others). One approach is to use approximate discrete Concrete distributions [24, 28], which admit sample-path gradients. However, samples from Concrete distributions are not discrete and significant approximation error can result in pathologies similar to evaluating the AF using continuous relaxation. Moreover, approximately discrete samples prohibit using surrogate models that require discrete inputs, e.g., GPs with Hamming distance kernels [37]. Another approach for gradient propagation in the deep learning community is to use straight-through gradient estimators (STE) [3], where the gradient of the discretization function with respect to its input is estimated using an identity function. This approach works well empirically, these estimators are not well-grounded theoretically. Nevertheless, we discuss and evaluate using STE for AF optimization in Appendix G.

6 Experiments

In this section, we provide an empirical evaluation of PR using SAA estimators on a suite of synthetic problems and real world applications. We use $N = 1,024$ MC samples in our experiments and demonstrate that PR is robust with respect to the number of MC samples (and compare against analytic PR, where computationally feasible) in Appendix F. We compare PR against two alternative acquisition optimization strategies: using a continuous relaxation (CONT. RELAX.) and using exact discretization and approximate gradients (EXACT ROUND) [20]. In addition, we compare against two state-of-the-art methods for discrete/mixed BO: a modified version of CASMOPOLITAN [45] that additionally supports ordinal variables introduced in Wan et al. [46] and HYBO [12], both of which are shown to outperform the other related works discussed in Section 5. In addition, we showcase how PR is complementary to existing methods such as trust region methods [17]. We demonstrate

this by using PR with a trust region for the continuous and discrete ordinal parameters and optimize PR within this trust region. In Appendix G, we provide comparison of TR methods with alterative optimizers and find that PR is the best optimizer for TR on 6 of the 7 benchmark problems. See Appendix C for additional discussion of PR + TR. For PR, EXACT ROUND, and PR + TR we use the sum of a product kernel and a sum kernel of a categorical kernel [37] for the categorical parameters and Matérn-5/2 kernel for all other parameters.[4]

CONT. RELAX., EXACT ROUND, PR, and PR + TR use expected improvement [19, 25] for single objective (constrained problems) and expected hypervolume improvement [16] for the multi-objective oil sorbent problem (where exact gradients with respect to continuous parameters are computed using auto-differentiation [7]). We report the mean for each method $\pm$ 2 standard errors across 20 replications. Performance is evaluated in terms of regret (feasible regret for constrained problems and hypervolume regret for multi-objective problems). CASMOPOLITAN and HYBO are not run on Welded Beam and Oil Sorbent as they do not support constrained and multi-objective optimization. We also leave the multi-objective extension of PR+TR to future work because it would add additional complexity [9]. For HYBO, we only run 60 BO iterations on SVM due to the large wall time (see Figure 3) and only report partial results on Cellular Network due to a singular covariance matrix error. See Appendix C for details on the experiment setup, regret metrics, benchmark problems, and methodological details. We leverage existing open source implementations of CASMOPOLITAN and HYBO (see Appendix C for links), and we will open source our implementations of all of other methods upon publication.

6.1 Synthetic Problems

We evaluate all methods on 3 synthetic problems: **Ackley** is a 13-dimensional function with 10 binary and 3 continuous parameters (a modified version of the problem in Bliek et al. [5]). **Mixed Int F1** is a 16-dimensional variant of the F1 function from Tušar et al. [44] with 2 binary, 6 discrete ordinal parameters, and 8 continuous parameters. The discrete ordinal parameters have following cardinalities: 2 parameters with 3 values, 2 with 5 values, and 2 with 7 values. **Rosenbrock** is a 10-dimensional Rosenbrock function with 6 discrete ordinal parameters with 4 values each and 4 continuous parameters.

6.2 Real World Problems

We consider 5 real world applications including a problem with 5 black-box outcome constraints and a 3-objective problem (see Appendix D for details on constrained and multi-objective BO).

Welded Beam Optimizing the design of a welded steel beam is a classical engineering optimization. In this problem, the goal is to minimize manufacturing cost subject to 5 black-box constraints on structural properties of the beam (including shear stress, bending stress, and buckling load) my tuning 6 parameters: the welding configuration (binary), the metal material type (categorical with 4 options), and 4 ordinal parameters controlling the dimensions of the beam [43].

SVM Feature Selection This problem involves jointly performing feature selection and hyperparameter optimization for a Support Vector Machine (SVM) trained on the CTSlice UCI data set [15, 27]. This design space for this problem involves 50 binary parameters controlling whether a particular feature is included or not, and 3 continuous hyperparameters of the SVM.

Cellular Network Optimization In this 30-dimensional problem, the goal is to tune the tilt (ordinal with 6 values) and transmission power (continuous) for a set of 15 antennas [39] to maximize a coverage quality metric that is a function of signal power and interference [29] over a geographic region of interest. We use the simulator from Dreifuerst et al. [14].

Direct Arylation Chemical Synthesis Palladium-catalysed direct arylation has generating significant interest in pharamceutical development [10]. In this problem, the goal is maximize yield for a direct arylation chemical reaction by tuning the 3 categorical parameters corresponding to the choice of solvent, base, and ligand, as well 2 continuous parameters controlling the temperature and concentration. We fit a surrogate model to the direct arylation dataset from Shields et al. [40] in order to facilitate continuous optimization of temperature and concentration.

Electrospun Oil Sorbent Marine oil spills can cause ecological catastrophe. One avenue for mitigating environmental harm is to design and deploy absorbent materials to capture the spilled oil. In

[4] CONT. RELAX. is incompatible with a categorical kernel so we use Matérn-5/2 with one-hot encoding.

this problem, we tune 5 ordinal parameters (3 parameters with 5 values and 2 with 4 values) and 2 continuous parameters controlling the composition and manufacturing conditions for an electrospun oil sorbent material to maximize 3 competing objectives: the oil absorbing capacity, the mechanical strength, and the water contact angle [47].

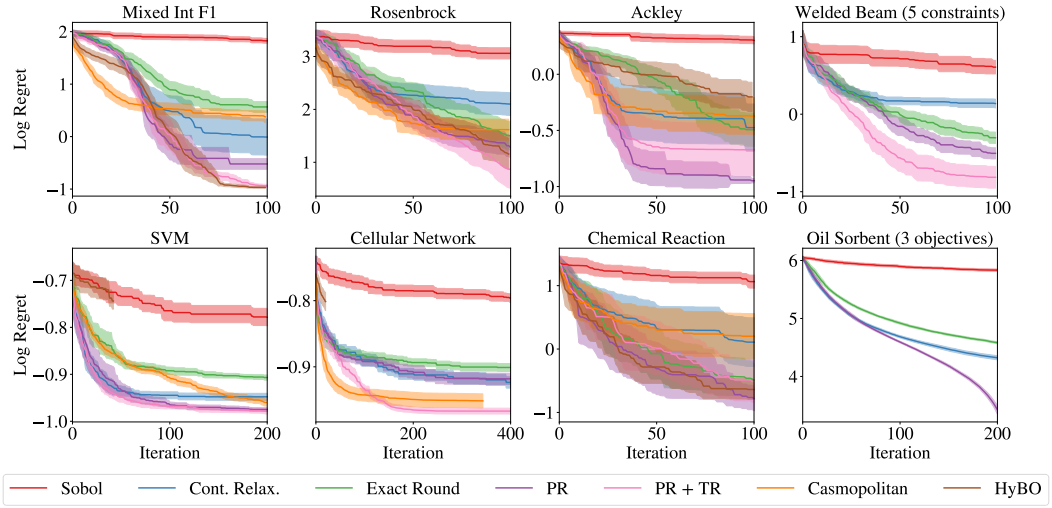

Figure 2: PR (or PR + TR) consistently outperforms alternatives with respect to log regret.

6.3 Results

We find PR consistently delivers strong empirical performance as shown in Figure 2. *On all benchmark problems,* PR *(or* PR + TR*) outperforms all baseline methods (except for Mixed Int F1, where* HYBO *performs comparably).* Figure 3 in Section E shows the wall time for candidate generation over the number of BO iterations. Although PR is computationally intensive, the computation is embarrassingly parallel and therefore exploiting GPU acceleration yields competitive wall times. Importantly, PR's wall time scales well with the number of observations and design parameters, unlike HYBO which scales poorly with both. However, the complexity of PR is multiplicative in the number of GPs being used (e.g. outcomes being modeled), assuming they are evaluated sequentially. Hence, in multi-objective or constrained settings, PR incurs a high cost in terms of wall time. However, empirically PR achieves better optimization performance on constrained and multi-objective problems relative to CONT. RELAX. and EXACT ROUND. We note that CASMOPOLITAN does not support multi-objective BO or constrained BO, and although HYBO could be used in those settings, it would be impractically slow because 1) its wall time would scale linearly with the number of modeled outcomes (using independent GPs) and 2) its diffusion kernel is non-differentiable, which would make optimizing hypervolume-based AFs slow [7, 8].

7 Discussion

The performance and regret properties of BO depend critically on adequately maximizing the AF. For problems with discrete features, exhaustively trying all possible combinations of discrete values quickly becomes infeasible as the number of combinations grows. Alternatives such as trying a subset of the possible combinations or resorting to continuous relaxations often leads to a failure to effectively optimizing the AF which may result in sub-optimal BO performance. As an alternative, we propose using PR which allows us to better optimize the AF and we showed that PR shows strong performance on a large number of real-world problems. Our approach is complementary to other approaches and can easily be combined with other approaches such as trust region methods and specialized kernels for discrete parameters. One limitation of PR is that it requires introduces computationally-demanding integration. However, given that the computation in PR is embarrassingly parallel, it motivates for future research on optimizing AFs on distributed hardware.

References

[1] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[2] Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *Proc. of ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 471–480. PMLR, 2018.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv preprint*, abs/1308.3432, 2013.

[4] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 2546–2554, 2011.

[5] Laurens Bliek, Arthur Guijt, Sicco Verwer, and Mathijs de Weerdt. Black-box mixed-variable optimisation using a surrogate model that satisfies integer constraints. GECCO '21, page 1851–1859. Association for Computing Machinery, 2021.

[6] Endre Boros and Peter L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1):155–225, 2002. ISSN 0166-218X. doi: https://doi.org/10.1016/S0166-218X(01)00341-9.

[7] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[8] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2187–2200. Curran Associates, Inc., 2021.

[9] Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. *ArXiv preprint*, abs/2109.10964, 2021.

[10] Huw ML Davies and Daniel Morton. Recent advances in c–h functionalization. *The Journal of Organic Chemistry*, 81(2):343–350, 2016.

[11] Erik A. Daxberger, Anastasia Makarova, Matteo Turchetta, and Andreas Krause. Mixed-variable bayesian optimization. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2633–2639. ijcai.org, 2020. doi: 10.24963/ijcai.2020/365.

[12] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 2632–2643. PMLR, 2021.

[13] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Mercer features for efficient combinatorial bayesian optimization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 7210–7218, 2021.

[14] Ryan M Dreifuerst, Samuel Daulton, Yuchen Qian, Paul Varkey, Maximilian Balandat, Sanjay Kasturia, Anoop Tomar, Ali Yazdan, Vish Ponnampalam, and Robert W Heath. Optimizing coverage and capacity in cellular networks using machine learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8138–8142. IEEE, 2021.

[15] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[16] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

[17] David Eriksson, Michael Pearce, Jacob R. Gardner, Ryan Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5497–5508, 2019.

[18] Peter I Frazier. A tutorial on bayesian optimization. *ArXiv preprint*, abs/1807.02811, 2018.

[19] Jacob R. Gardner, Matt J. Kusner, Zhixiang Eddie Xu, Kilian Q. Weinberger, and John P. Cunningham. Bayesian optimization with inequality constraints. In *Proc. of ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 937–945. JMLR.org, 2014.

[20] Eduardo C. Garrido-Merchán and D. Hernández-Lobato. Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35, 2020.

[21] Peter W Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

[22] Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, and Alán Aspuru-Guzik. Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews*, 8(3):031406, 2021.

[23] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the 5th International Conference on Learning and Intelligent Optimization*, page 507–523. Springer-Verlag, 2011. ISBN 9783642255656.

[24] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *Proc. of ICLR*. OpenReview.net, 2017.

[25] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[26] Jack PC Kleijnen and Reuven Y Rubinstein. Optimization and sensitivity analysis of computer simulation models by the score function method. *European Journal of Operational Research*, 88(3):413–427, 1996.

[27] Sulin Liu, Qing Feng, David Eriksson, Benjamin Letham, and Eytan Bakshy. Sparse bayesian optimization. *arXiv preprint arXiv:2203.01900*, 2022.

[28] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proc. of ICLR*. OpenReview.net, 2017.

[29] Wesley J Maddox, Maximilian Balandat, Andrew G Wilson, and Eytan Bakshy. Bayesian optimization with high-dimensional outputs. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[30] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21(132):1–62, 2020.

[31] Dang Nguyen, Sunil Gupta, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Bayesian optimization for categorical and category-specific continuous inputs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5256–5263. AAAI Press, 2020.

[32] ChangYong Oh, Jakub M. Tomczak, Efstratios Gavves, and Max Welling. Combinatorial bayesian optimization using the graph cartesian product. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2910–2920, 2019.

[33] Art B Owen. Quasi-monte carlo sampling. *Monte Carlo Ray Tracing: Siggraph*, 1:69–88, 2003.

[34] Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 766–776. AUAI Press, 2019.

[35] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, 2004.

[36] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. ISSN 00034851.

[37] Bin Xin Ru, Ahsan S. Alvi, Vu Nguyen, Michael A. Osborne, and Stephen J. Roberts. Bayesian optimisation over multiple continuous and categorical inputs. In *Proc. of ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 8276–8285. PMLR, 2020.

[38] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *arXiv preprint: arXiv 1301.2609*, 2014.

[39] Soumya Ranjan Samal, Kaliprasanna Swain, Shuvabrata Bandopadhaya, Nikolay Dandanov, Vladimir Poulkov, Sidheswar Routray, and Gopinath Palai. Dynamic coverage optimization for 5g ultra-dense cellular networks based on their user densities. 2021.

[40] Benjamin J Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I Martinez Alvarado, Jacob M Janey, Ryan P Adams, and Abigail G Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021.

[41] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proc. of ICML*, pages 1015–1022. Omnipress, 2010.

[42] The GPyOpt authors. GPyOpt: A bayesian optimization framework in python. http://github.com/SheffieldML/GPyOpt, 2016.

[43] Anh Tran, Minh Tran, and Yan Wang. Constrained mixed-integer gaussian mixture bayesian optimization and its applications in designing fractal and auxetic metamaterials. *Structural and Multidisciplinary Optimization*, 59(6):2131–2154, 2019.

[44] Tea Tušar, Dimo Brockhoff, and Nikolaus Hansen. Mixed-integer benchmark problems for single- and bi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, page 718–726. Association for Computing Machinery, 2019.

[45] Xingchen Wan, Vu Nguyen, Huong Ha, Bin Xin Ru, Cong Lu, and Michael A. Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 10663–10674. PMLR, 2021.

[46] Xingchen Wan, Cong Lu, Jack Parker-Holder, Philip J Ball, Vu Nguyen, Binxin Ru, and Michael Osborne. Bayesian generational population-based training. In *ICLR Workshop on Agent Learning in Open-Endedness*, 2022.

[47] Boqian Wang, Jiacheng Cai, Chuangui Liu, Jian Yang, and Xianting Ding. Harnessing a novel machine-learning-assisted evolutionary algorithm to co-optimize three characteristics of an electrospun oil sorbent. *ACS Applied Materials & Interfaces*, 12(38):42842–42849, 2020.

[48] Jialei Wang, Scott C Clark, Eric Liu, and Peter I Frazier. Parallel bayesian global optimization of expensive functions. *Operations Research*, 68(6):1850–1865, 2020.

[49] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[50] James T. Wilson, Frank Hutter, and Marc Peter Deisenroth. Maximizing acquisition functions for bayesian optimization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9906–9917, 2018.

[51] Mingzhang Yin, Yuguang Yue, and Mingyuan Zhou. ARSM: augment-reinforce-swap-merge estimator for gradient backpropagation through categorical variables. In *Proc. of ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 7095–7104. PMLR, 2019.

[52] Mingzhang Yin, Nhat Ho, Bowei Yan, Xiaoning Qian, and Mingyuan Zhou. Probabilistic Best Subset Selection by Gradient-Based Optimization. *arXiv e-prints*, 2020.

[53] Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *Proc. of ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 11096–11105. PMLR, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix A.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix C.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] See Appendix C.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]