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 reparameterizations, 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 converges to a stationary point of the probabilistic objective under gradient ascent using scalable, unbiased estimators of both the probabilistic objective and its gradient, and therefore, as the numbers of starting points and gradient steps increase our approach will recover of a maximizer of the AF (an often neglected requisite for commonly-used BO regret bounds). 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 [57]. 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 [53]. 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.

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

Bayesian optimization (BO) is a popular technique for sample-efficient black-box optimization, due to its proven performance guarantees in many settings [5, 51] and its strong empirical performance [23]. BO leverages a probabilistic surrogate model of the unknown objective(s) and an acquisition function (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 [51]. 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 there is an equivalence between the maximizers of the acquisition function and the 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 PO 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 and gradient steps increase. This is important because many BO regret bounds require maximizing the AF [51]. Although the AF is often non-convex and maximization is hard, empirically, with a modest number of starting points, PR leads to better AF optimization than alternative methods.
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) [45]—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) [32] and upper confidence bound (UCB) [51]—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).

[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.

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 [39, 47] and local search [40] for discrete domains and interleaved discrete/continuous optimization procedures for mixed domains [15, 55]. A simple and widely-used approach across many popular BO packages [1, 52] 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	$z \in \{0, ..., C-1\}$	$z' \in [-0.5, C-0.5)$	$\text{round}(z')$
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 AF 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 AF 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 AF value, but discretization will result in a design that has already been evaluated and has zero AF value. To mitigate this degenerate behavior and avoid the over-estimation issue, Garrido-Merchán and Hernández-Lobato [25] propose discretizing z' before evaluating the AF, but the AF is then non-differentiable with respect to the z' . While this improves performance on small search spaces, the AF and surrogate have large flat regions after discretizing z', which makes it difficult to optimize the AF. The authors of [25] propose to approximate the gradients using finite differences, but, empirically, we find that this approach to be sub-optimal 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. We denote the $C-1$-simplex as Δ^{C-1}.

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	$Z \sim \text{CATEGORICAL}(\theta), \theta = (\theta^{(1)}, ..., \theta^{(C)})$	$\theta \in \Delta^{C-1}$

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, maximizing the PO allows us to recover a maximizer of α over the space $\mathcal{X} \times \mathcal{Z}$. 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 $p(\boldsymbol{Z}|\boldsymbol{\theta})$ is a discrete probability distribution, we

can express $\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$ as a linear combination where each discrete design is weighted by its probability mass:

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

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

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

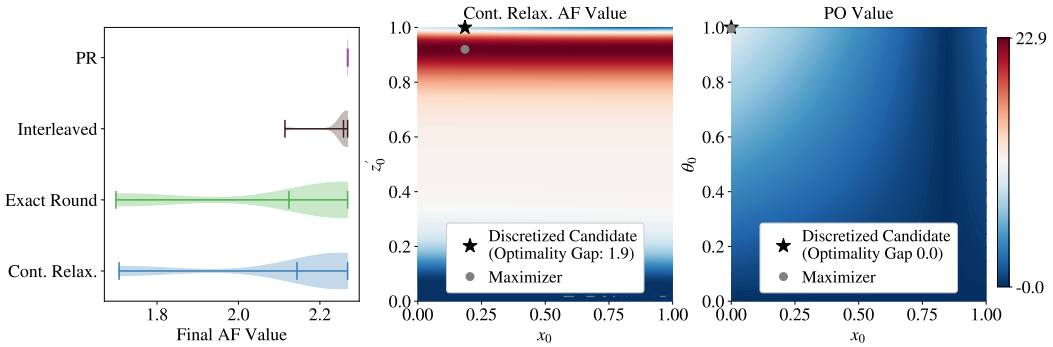

Figure 1: (**Left**) A comparison of AF optimization using different methods over a mixed search space shows that PR *outperforms alternative methods for AF optimization and has much lower variance across replications*. For each method, the best candidate across 20 restarts is selected (after discretization) 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. Proofs are provided in Appendix B. Our first result is that there is an equivalence between the maximizers of the PO and the maximizers of the AF over $\mathcal{X} \times \mathcal{Z}$.

Theorem 1 (Consistent Maximizers). *Suppose that α is continuous in $\boldsymbol{x}$ for every $\boldsymbol{z} \in \mathcal{Z}$. 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 defined as: $\hat{\mathcal{H}}^* = \{(\boldsymbol{x}, \tilde{\boldsymbol{z}}) : (\boldsymbol{x}, \boldsymbol{\theta}) \in \mathcal{J}^*, \tilde{\boldsymbol{z}} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})\}$. Then, $\hat{\mathcal{H}}^* = \mathcal{H}^*$.*

Algorithm 1 outlines BO with probabilistic reparameterization. Importantly, Theorem 1 states that sampling from the distribution parameterized by a maximizer of the PO yields a maximizer of α, and therefore, 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.*

Algorithm 1 BO with PR

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

Examples of such AFs are upper confidence bound (UCB) [51] or Thompson sampling (TS) [48] for single objective optimization, and UCB or TS with Chebyshev [42] or hypervolume [64] scalarizations in the multi-objective setting.

Although the BO policy selects a maximizer of α is equivalent to the BO policy in Algorithm 1, 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 or other common relaxations as shown in Figure 1(Left), where we compare AF optimization methods on the chemical synthesis problem [50] (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 [25]. "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 [55]). All methods except interleaved use L-BFGS-B and the expected improvement AF [32].

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 [34] (also known as REINFORCE [60]) and the likelihood ratio estimator [26]) 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. [62], Yin et al. [63]) would require many additional AF evaluations (see Mohamed et al. [38] 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 [62, 63]. 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 [46] to arrive at our main convergence result for acquisition optimization.

Theorem 2 (Convergence Guarantee). *Let $\alpha : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ be differentiable in $\boldsymbol{x}$ for every $\boldsymbol{z} \in \mathcal{Z}$. 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 $\hat{\boldsymbol{z}}_{t,m} \sim p(\boldsymbol{Z}|\hat{\boldsymbol{\theta}}_{t,m})$. Then as $t \to \infty$ and $m \to \infty$, $(\hat{\boldsymbol{x}}_{t,m}, \hat{\boldsymbol{z}}_{t,m}) \to (\boldsymbol{x}^*, \boldsymbol{z}^*) \in \arg\max_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{X}\times\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})$ in probability.*

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 to 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.4. 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 [41] instead of i.i.d. 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 [30] 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 [40]. MERCBO similarly exploits the combinatorial graph, but with Mercer features and Thompson sampling [16]. HYBO extends the diffusion kernels to mixed continuous-discrete spaces [15]. 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 [28] uses kernel density estimation, but is limited to categorical search spaces. MIVABO uses a linear combination of basis functions (e.g. pseudo Boolean features [7] for discrete parameters) with interaction terms[13]. MVRSM [6] 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 [25] 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 [47] 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 [55] 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 [60] and sparse regression [63]. However, PR has not been leveraged for BO (although a different reparameterization is commonly used in MC AFs [61]).

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 [31, 36], 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 [47]. 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) [25]. In addition, we compare against two state-of-the-art methods for discrete/mixed BO: a modified version of CASMOPOLITAN [55] that additionally supports ordinal variables introduced in Wan et al. [56] and HYBO [15], 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 [22]. 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 [47] for the categorical parameters and Matérn-5/2 kernel for all other parameters.[3]

CONT. RELAX., EXACT ROUND, PR, and PR + TR use expected improvement [24, 32] for single objective (constrained problems) and expected hypervolume improvement [20] for the multi-objective oil sorbent problem (where exact gradients with respect to continuous parameters are computed using auto-differentiation [8]). 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 [11]. For HYBO, we only run 60 BO iterations on SVM due to the large wall time (see Figure 4) 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. [6]). **Mixed Int F1** is a 16-dimensional variant of the F1 function from Tušar et al. [54] 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 [53].

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 [18, 35]. 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 [49] to maximize a coverage quality metric that is a function of signal power and interference [37] over a geographic region of interest. We use the simulator from Dreifuerst et al. [17].

Direct Arylation Chemical Synthesis Palladium-catalysed direct arylation has generating significant interest in pharamceutical development [12]. 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. [50] 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

[3] 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 [57].

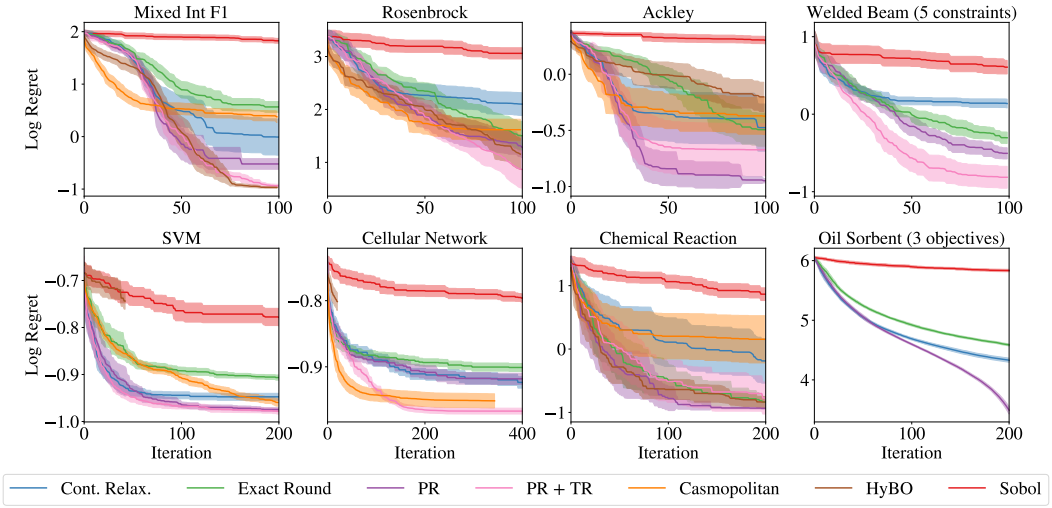

Figure 2: PR (or PR + TR) consistently outperforms alternatives with respect to log regret.

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 4 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 [8, 9].

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] Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24, 2019. URL `http://jmlr.org/papers/v20/18-213.html`.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] Samuel Daulton, Sait Cakmak, Maximilian Balandat, Michael A Osborne, Enlu Zhou, and Eytan Bakshy. Robust multi-objective bayesian optimization under input noise. *arXiv preprint arXiv:2202.07549*, 2022.

[11] Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, Eindhoven, Netherlands, August 1-5, 2022*, Proceedings of Machine Learning Research. AUAI Press, 2022.

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

[13] 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.

[14] Aryan Deshwal and Jana Doppa. Combining latent space and structured kernels for bayesian optimization over combinatorial spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8185–8200. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/44e76e99b5e194377e955b13fb12f630-Paper.pdf`.

[15] 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.

[16] 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.

[17] 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.

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

[19] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Comb. Probab. Comput.*, 13(4–5): 577–625, jul 2004. ISSN 0963-5483. doi: 10.1017/S0963548304006315.

[20] 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.

[21] David Eriksson and Matthias Poloczek. Scalable constrained Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 730–738. PMLR, 2021.

[22] 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.

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

[24] 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.

[25] 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.

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

[27] Nikolaus Hansen, Dimo Brockhoff, Olaf Mersmann, Tea Tusar, Dejan Tusar, Ouassim Ait ElHara, Phillipe R Sampaio, Asma Atamna, Konstantinos Varelas, Umut Batu, et al. Comparing continuous optimizers: numbbo/coco on github. 2019.

[28] 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.

[29] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[30] 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.

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

[32] 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.

[33] Kirthevasan Kandasamy, Jeff Schneider, and Barnabas Poczos. High dimensional bayesian optimisation and bandits via additive models. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 295–304, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/kandasamy15.html`.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

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

[42] 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.

[43] Julien Pelamatti, Loïc Brevault, Mathieu Balesdent, El-Ghazali Talbi, and Yannick Guerin. Bayesian optimization of variable-size design space problems. *Optimization and Engineering*, 22(1):387–447, 2021.

[44] J. Rapin and O. Teytaud. Nevergrad - A gradient-free optimization platform. `https://GitHub.com/FacebookResearch/Nevergrad`, 2018.

[45] 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.

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

[47] 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.

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

[49] 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.

[50] 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.

[51] 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.

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

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] Rui Wang, Jian Xiong, Hisao Ishibuchi, Guohua Wu, and Tao Zhang. On the effect of reference point in moea/d for multi-objective optimization. *Applied Soft Computing*, 58:25–34, 2017. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2017.04.002.

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

[61] 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.

[62] 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.

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

[64] 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.

[65] Yichi Zhang, Siyu Tao, Wei Chen, and Daniel Apley. A latent variable approach to gaussian process modeling with qualitative and quantitative factors. *Technometrics*, 62:1–19, 07 2019. doi: 10.1080/00401706.2019.1638834.

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]

Appendix to:

Bayesian Optimization over Discrete and Mixed Spaces via Probabilistic Reparameterization

A Potential Societal Impacts

Our work advances Bayesian optimization, a generic class of methods for optimization of expensive, difficult-to-optimize black-box problems. With this paper in particular, we improve the performance of Bayesian optimization on problems with mixed types of inputs. Given the ubiquity of such problems in many practical applications, we believe that our method could lead to positive broader impacts by solving these problems better and more efficiently while reducing the costs incurred for solving them. Concrete and high-stake examples where our method could be potentially applied (some of which have been already demonstrated by the benchmark problems considered in the paper) include but are not limited to applications in communications, chemical synthesis, drug discovery, engineering optimization, tuning of recommender systems, and automation of machine learning systems. On the flip side, while the method proposed is ethically neutral, there is potential of misuse given that the exact objective of optimization is ultimately decided by the end users; we believe that practitioners and researchers should be aware of such possibility and aim to mitigate any potential negative impacts to the furthest extent.

B Theoretical Results and Proofs

B.1 Results

Let $\mathcal{P}_{\mathcal{Z}}^{(i)} := \mathcal{P}(\mathcal{Z}^{(i)})$ denote the set of probability measures on $\mathcal{Z}^{(i)}$ for each $i = 1, ..., d_z$, and let $\mathcal{P}_{\mathcal{Z}} := \prod_{i=1}^{d_z} \mathcal{P}_{\mathcal{Z}}^{(i)}$. For any $\alpha : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$, define $\tilde{\alpha} : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$ as

$$\tilde{\alpha}(\boldsymbol{x}, p) = \int_{\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) dp(\boldsymbol{z}) = \sum_{\boldsymbol{z} \in \mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) p(\{\boldsymbol{z}\}). \tag{8}$$

Let Θ be a compact metric space, and consider the set of functionals $\Phi = \{\varphi \; s.t. \; \varphi : \Theta \to \mathcal{P}_{\mathcal{Z}}\}$. Let

$$\hat{\alpha}(\boldsymbol{x}, \boldsymbol{\theta}) := \tilde{\alpha}(\boldsymbol{x}, \boldsymbol{\varphi}(\boldsymbol{\theta})) = \int_{\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) dp_{\varphi(\boldsymbol{\theta})}(\boldsymbol{z}) = \sum_{\boldsymbol{z} \in \mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) p_{\varphi(\boldsymbol{\theta})}(\{\boldsymbol{z}\}) \tag{9}$$

Since $\mathcal{Z}$ is finite, each element of $\varphi \in \Phi$ can be expressed as a mapping from Θ to $\mathbb{R}^{|\mathcal{Z}|}$. Namely, each $\varphi(\boldsymbol{\theta})$ corresponds to a vector with $|\mathcal{Z}|$ elements containing the probability mass for each element of $\mathcal{Z}$ under $p_{\varphi}(\boldsymbol{\theta})$. Thus $(\mathcal{P}_{\mathcal{Z}}, \|\cdot\|)$ is a metric space under any norm $\|\cdot\|$ on $\mathbb{R}^{|\mathcal{Z}|}$. Let $\alpha^* := \max_{(\boldsymbol{x},\boldsymbol{z}) \in (\mathcal{X} \times \mathcal{Z})} \alpha(\boldsymbol{x}, \boldsymbol{z})$ and let $\mathcal{H}^* := \arg\max_{(\boldsymbol{x},\boldsymbol{z}) \in (\mathcal{X} \times \mathcal{Z})} \alpha(\boldsymbol{x}, \boldsymbol{z})$ denote the set of maximizers of α.

Lemma 1. *Suppose α is continuous in $\boldsymbol{x}$ for every $\boldsymbol{z} \in \mathcal{Z}$ and that $\varphi : \Theta \mapsto (\mathcal{P}_{\mathcal{Z}}, \|\cdot\|)$ is continuous with $\varphi(\Theta) = \mathcal{P}_{\mathcal{Z}}$. Let $\mathcal{J}^* := \arg\max_{(\boldsymbol{x},\boldsymbol{\theta}) \in \mathcal{X} \times \Theta} \hat{\alpha}(\boldsymbol{x}, \boldsymbol{\theta})$. Then for any $(\boldsymbol{x}^*, \boldsymbol{\theta}^*) \in \mathcal{J}^*$, it holds that $(\boldsymbol{x}^*, \boldsymbol{z}) \in \mathcal{H}^*$ for all $z \in \operatorname{supp} p_{\varphi(\boldsymbol{\theta}^*)}$.*

Proof. First, note that $\hat{\alpha} : \mathcal{X} \times \Theta \to \mathbb{R}$ is continuous (using that φ is continuous and α is bounded). Since both $\mathcal{X}$ and Θ are compact $\hat{\alpha}$ attains its maximum, i.e., $\mathcal{J}^*$ exists. Let $(\boldsymbol{x}^*, \boldsymbol{\theta}^*) \in \mathcal{J}^*$. Clearly, there exists $\boldsymbol{z}^* \in \arg\max_{\boldsymbol{z} \in \mathcal{Z}} \alpha(\boldsymbol{x}^*, \boldsymbol{z})$ such that $\alpha(\boldsymbol{x}^*, \boldsymbol{z}^*) = \alpha^*$. Suppose there exists $\boldsymbol{z}' \in \operatorname{supp} p_{\varphi(\boldsymbol{\theta}^*)}$ such that $(\boldsymbol{x}^*, \boldsymbol{z}') \notin \mathcal{H}^*$. Then $\alpha(\boldsymbol{x}^*, \boldsymbol{z}') < \alpha^*$ and, since $\mathcal{Z}$ is finite, $p_{\varphi(\boldsymbol{\theta}^*)}(\{\boldsymbol{z}'\}) > 0$.

Consider the probability measure p' given by

$$p'(\{\boldsymbol{z}\}) = \begin{cases} 0 & \text{if } \boldsymbol{z} = \boldsymbol{z}' \\ p_{\varphi(\boldsymbol{\theta}^*)}(\{\boldsymbol{z}^*\}) + p_{\varphi(\boldsymbol{\theta}^*)}(\{\boldsymbol{z}'\}) & \text{if } \boldsymbol{z} = \boldsymbol{z}^* \\ p_{\varphi(\boldsymbol{\theta}^*)}(\{\boldsymbol{z}\}) & \text{otherwise} \end{cases}$$

Then

$$\begin{aligned} \tilde{\alpha}(\boldsymbol{x}^*, p') - \hat{\alpha}(\boldsymbol{x}^*, \boldsymbol{\theta}^*) &= \sum_{\boldsymbol{z} \in \mathcal{Z}} \alpha(\boldsymbol{x}^*, z) p'(\{\boldsymbol{z}\}) - \hat{\alpha}(\boldsymbol{x}^*, \boldsymbol{\theta}^*) \\ &= \sum_{\boldsymbol{z} \in \mathcal{Z}} \alpha(\boldsymbol{x}^*, \boldsymbol{z}) p_{\varphi(\boldsymbol{\theta}^*)}(\{\boldsymbol{z}\}) + p_{\varphi(\boldsymbol{\theta}^*)}(\{\boldsymbol{z}'\})(\alpha(\boldsymbol{x}^*, \boldsymbol{z}^*) - \alpha(\boldsymbol{x}^*, \boldsymbol{z}')) \\ &\quad - \hat{\alpha}(\boldsymbol{x}^*, \boldsymbol{\theta}^*) \\ &= p_{\varphi(\boldsymbol{\theta}^*)}(\{\boldsymbol{z}'\})(\alpha(\boldsymbol{x}^*, \boldsymbol{z}^*) - \alpha(\boldsymbol{x}^*, \boldsymbol{z}')) \\ &> 0 \end{aligned}$$

Now $p' \in \mathcal{P}_{\mathcal{Z}}$, and so $p' = \varphi(\boldsymbol{\theta}')$ for some $\boldsymbol{\theta}' \in \Theta$. But then $\hat{\alpha}(\boldsymbol{x}^*, \boldsymbol{\theta}') > \hat{\alpha}(\boldsymbol{x}^*, \boldsymbol{\theta}^*)$. This is a contradiction. □

Corollary 2. *Suppose the optimizer of g is unique, i.e., that $\mathcal{H}^* = \{(\boldsymbol{x}^*, \boldsymbol{z}^*)\}$ is a singleton. Then the optimizer of $\hat{\alpha}$ is also unique and $\mathcal{J}^* = \{(\boldsymbol{x}^*, \boldsymbol{\theta}^*)\}$, with $p_{\varphi(\boldsymbol{\theta}^*)}(\{\boldsymbol{z}^*\}) = 1$.*

Corollary 3. *Consider the following mappings:*

- ***Binary:*** *$\varphi : [0, 1] \to \mathcal{P}_{\{0,1\}}$ with $p_{\varphi(\theta)}(\{1\}) = \theta$ and $p_{\varphi(\theta)}(\{0\}) = 1 - \theta$.*
- ***Ordinal:*** *$\varphi : [0, C-1] \to \mathcal{P}_{\{0,1,\ldots,C\}}$ with $p_{\varphi(\theta)}(\{i\}) = (1 - |i - \theta|)\, \mathbf{1}\{|i - \theta| \leq 1\}$ for $i = 1, \ldots, C$.*
- ***Categorical:*** *$\varphi : [0, 1]^C \to \mathcal{P}_{\{0,1,\ldots,C\}}$ with $p_{\varphi(\theta)}(\{i\}) = \frac{\theta_i}{\sum_{i=1}^{C} \theta_i}$.*

These mappings satisfy the conditions for Lemma 1. In the setting with multiple discrete parameters where the above mappings are applied in component-wise fashion for each discrete parameter, the component-wise mappings also satisfy the conditions for Lemma 1.

Clearly, the mappings given in Corollary 3 are continuous functions of θ. In the setting with multiple discrete parameters , the component-wise function is also continuous with respect to the distribution parameters for each discrete parameter. Hence, the mappings satisfy the conditions for Lemma 1.

Lemma 2. *If $(\boldsymbol{x}^*, \boldsymbol{z}^*) \in \mathcal{H}^* = \arg\max_{(\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{X} \times \mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})$, then*

$$\alpha(\boldsymbol{x}^*, \boldsymbol{z}^*) = \max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}^*, \boldsymbol{Z})].$$

Proof. For any $\boldsymbol{z}^*$, let $\boldsymbol{\theta}^*$ be the parameters such that $p(\boldsymbol{z}^*|\boldsymbol{\theta}^*) = 1$ (i.e. a point mass on $\boldsymbol{z}^*$). From Equation (2),

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

Claim: $\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta}^*)}[\alpha(\boldsymbol{x}^*, \boldsymbol{Z})] = \max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}^*, \boldsymbol{Z})]$.

Suppose there exists $\boldsymbol{\theta}'$ such that $\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta}')}[\alpha(\boldsymbol{x}^*, \boldsymbol{Z})] > \mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta}^*)}[\alpha(\boldsymbol{x}^*, \boldsymbol{Z})]$. Since $(\boldsymbol{x}^*, \boldsymbol{z}^*) \in \mathcal{H}^*$, $\alpha(\boldsymbol{x}^*, \boldsymbol{z}^*) = \max_{(\boldsymbol{x}, \boldsymbol{z}) \in \mathcal{X} \times \mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})$. Hence, there is no convex combination of values of α that is greater than $\alpha(\boldsymbol{x}^*, \boldsymbol{z}^*)$. This is a contradiction. □

Theorem 1 (Consistent Maximizers)**.** *Suppose that α is continuous in $\boldsymbol{x}$ for every $\boldsymbol{z} \in \mathcal{Z}$. 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 defined as: $\hat{\mathcal{H}}^* = \{(\boldsymbol{x}, \tilde{\boldsymbol{z}}) : (\boldsymbol{x}, \boldsymbol{\theta}) \in \mathcal{J}^*, \tilde{\boldsymbol{z}} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})\}$. Then, $\hat{\mathcal{H}}^* = \mathcal{H}^*$.*

Proof. From Lemma 1, we have that for any $(\boldsymbol{x}^*, \boldsymbol{\theta}^*) \in \mathcal{J}^*$, it holds that $(\boldsymbol{x}^*, \boldsymbol{z}) \in \mathcal{H}^*$ for all $z \in \operatorname{supp} p_{\varphi(\boldsymbol{\theta}^*)}$. Hence, $\hat{\mathcal{H}}^* \subseteq \mathcal{H}^*$.

Now, let $(\boldsymbol{x}^*, \boldsymbol{z}^*) \in \mathcal{H}^*$. Let $\boldsymbol{\theta}^* \in \Theta$ such that $p(\boldsymbol{z}^*|\boldsymbol{\theta}^*) = 1$. From the proof of Lemma 2, we have that $\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta}^*)}[\alpha(\boldsymbol{x}^*, \boldsymbol{Z})] = \alpha(\boldsymbol{x}^*, \boldsymbol{z}^*)$. As in the proof of Lemma 2, there is no convex combination of values of α greater than $\alpha(\boldsymbol{x}^*, \boldsymbol{z}^*)$. So $\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta}^*)}[\alpha(\boldsymbol{x}^*, \boldsymbol{Z})] = \max_{(\boldsymbol{x},\boldsymbol{\theta})\in\mathcal{X}\times\Theta} \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$, and therefore, $\boldsymbol{x}^*, \boldsymbol{\theta}^* \in \mathcal{J}^*$. Hence $(\boldsymbol{x}^*, \boldsymbol{z}^*) \in \hat{\mathcal{H}}^*$. So $\mathcal{H}^* \subseteq \hat{\mathcal{H}}^*$, and hence, $\hat{\mathcal{H}}^* = \mathcal{H}^*$. □

Lemma 3. *Suppose that $\alpha : (\boldsymbol{x}, \boldsymbol{z}) \mapsto \mathbb{R}$ is differentiable with respect to $\boldsymbol{x}$ for all $\boldsymbol{z} \in \mathcal{Z}$, and that the mapping $\varphi : \boldsymbol{\theta} \mapsto \mathcal{P}_{\mathcal{Z}}$ is such that $p_{\varphi(\boldsymbol{\theta})}(\{\boldsymbol{z}\})$ is differentiable with respect to $\boldsymbol{\theta}$ for all $z \in \mathcal{Z}$. Then the probabilistic objective $\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]$ is differentiable with respect to $(\boldsymbol{x}, \boldsymbol{\theta})$.*

Proof. For any $\boldsymbol{z} \in \mathcal{Z}$, the function $p(\boldsymbol{z}, \boldsymbol{\theta})\alpha(\boldsymbol{x}, \boldsymbol{z}) = p_{\varphi(\boldsymbol{\theta})}(\{\boldsymbol{z}\})\alpha(\boldsymbol{x}, \boldsymbol{z})$ is the product of two differentiable functions, hence differentiable. Therefore the probabilistic objective is a (finite) linear combination of differentiable functions, hence differentiable. □

Theorem 2 (Convergence Guarantee)**.** *Let $\alpha : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ be differentiable in $\boldsymbol{x}$ for every $\boldsymbol{z} \in \mathcal{Z}$. 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 $\hat{\boldsymbol{z}}_{t,m} \sim p(\boldsymbol{Z}|\hat{\boldsymbol{\theta}}_{t,m})$. Then as $t \to \infty$ and $m \to \infty$, $(\hat{\boldsymbol{x}}_{t,m}, \hat{\boldsymbol{z}}_{t,m}) \to (\boldsymbol{x}^*, \boldsymbol{z}^*) \in \arg\max_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{X}\times\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})$ in probability.*

Proof. The binary and categorical mappings in Corollary 3 are differentiable in θ (the ordinal mapping is differentiable almost everywhere[4]). Since the acquisition function $\alpha : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ is differentiable in $\boldsymbol{x}$ for every $\boldsymbol{z} \in \mathcal{Z}$, this means that the PO is differentiable. Using the prescribed sequence of step sizes, optimizing the PO using stochastic gradient ascent will converge almost surely to a local maximum after a sufficient number of steps [46]. As we increase the number of randomly distributed starting points, the probability of not finding the global maximum of the PO will converge to zero [58]. From Theorem 1, the PO and the AF have the same set of maximizers. Hence, convergence in probability to a global maximizer of the PO means convergence in probability to a global maximizer of the AF. □

C Experiment Details

For each BO optimization replicate, we use $N_{\text{init}} = \min(20, 2 * d_{\text{eff}})$ points from a scrambled sobol sequence, where d_{eff} is the "effective dimensionality" after one-hot encoding categorical parameters. Unless otherwise noted, all experiments use 20 replications and confidence intervals represent 2 standard errors of the mean. The same initial points are used for all methods for that replicate and different initial points are used for each replicate. For each method we report the $\log_{10}$ regret. Since the optimal value is unknown for many problems, we set the optimal value to be $\max(1.001 \cdot f^*, f^* + 0.1)$ where f^* is the best observed value across all methods and all replications. For constrained optimization f^* is the best feasible observed value and for multi-objective optimization f^* is the maximum hypervolume across all methods and replications. In total, the experiments in the main text (excluding HyBO and Casmopolitan) ran for an equivalent of 2,009.82 hours on a single Tesla V100-SXM2-16GB GPU. The baseline experiments (HyBO and Casmopolitan) ran for an equivalent of 745.10 hours on a single Intel Xeon Gold 6252N CPU.

[4]Technically, the arguments presented here do not prove convergence under the ordinal mapping, but we have found this to work well and reliably in practice. Alternatively, ordinal parameters could also just be treated as categorical ones in which case the convergence results hold. In practice, however, this introduces additional optimization variables that make the problem unnecessarily hard by removing the ordered structure from the problem.

C.1 Additional Problem Details

In this section, we describe the details of each synthetic problem considered in the experiments (the details of the remaining real-world problems are already described in Section 6.2).

Ackley. We use an adapted version of the 13-dimensional Ackley function modified from Bliek et al. [6]. The function is given by:

$$f(\mathbf{x}) = -a \exp\Big(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2)}\Big) - \exp\Big(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\Big) + a + \exp(1), \tag{10}$$

where in this case $a = 20, n = 0.2, c = 2\pi$ and $d = 13$ and $\mathbf{x} \in [-1, 1]^{13}$. We discretize the first 10 dimensions to be binary with the choice $\{-1, 1\}$, and the final 3 dimensions are unmodified with the original bounds.

Mixed Int F1. Mixed Int F1 is a partially discretized version of the 16-dimensional Sphere optimization problem [27], given by:

$$f(\mathbf{x}) = \sum_{i=1}^{d}(x_i - x_{\text{opt},i})^2 + f_{\text{opt}}, \tag{11}$$

where f_{opt} is sampled from a Cauchy distribution with median = 0 and roughly 50% of the values between -100 and 100. The sampled f_{opt} is then clamped to be between $[-1000, 1000]$ and rounded to the nearest integer. $\mathbf{x}_{\text{opt}}$ is sampled uniformly in $[-4, 4]^d$, and in this case $d = 16$. We discretize the first 8 dimensions as follows: the first 2 dimensions are binary with 2 choices $\{-5, 5\}$; the next 2 dimensions are ordinal with 3 choices $\{-5, 0, 5\}$; the next 2 dimensions are ordinal with 5 choices $\{-5, -2.5, 0, 2.5, 5\}$; the final 2 dimensions are ordinal with 7 choices $\{-5, -\frac{10}{3}, -\frac{5}{3}, 0, \frac{5}{3}, \frac{10}{3}, 5\}$. The remaining 8 dimensions are continuous with bounds $[-5, 5]^8$.

Rosenbrock. We use an adapted version of the Rosenbrock function, given by:

$$f(\mathbf{x}) = \Big(\sum_{i=1}^{d-1}\big(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\big)\Big), \tag{12}$$

where in this case $d = 10$. The first 6 dimensions are discretized to be ordinal variables, with 4 possible values each $x_i \in \{-5, 0, 5, 10\} \forall i \in [1, 6]$. The final 4 dimensions are continuous with bounds $[-5, 10]^4$.

Chemical Reaction (Direct Arylation Chemical Synthesis). For this problem, we fit a GP surrogate (with the same kernel used by the BO methods) to the dataset from Shields et al. [50] (available at `https://github.com/b-shields/edbo/tree/master/experiments/data/direct_arylation` under the MIT license) in order to facilitate continuous optimization of temperature and concentration. The surrogate is included with our source code.

Oil Sorbent. We set the reference point for this problem to be $[-125.3865, -57.8292, 43.2665]$, which we choose using a commonly used heuristic to scale the nadir point (component-wise worst objective values across the Pareto frontier) [59].

C.2 Method details

PR, Cont. Relax., Exact Round, PR + TR, and Exact Round + STE. We implemented all of these methods using BoTorch [1], which is available under the MIT license at `https://github.com/pytorch/botorch`. All AFs are optimized via L-BFGS-B with 20 random restarts, run for a maximum of 200 iterations. We follow the default initialization heuristic in BoTorch [1], which initializes the optimizer by evaluating the acquisition function at a large number of starting points (here, 1024, chosen from a scrambled Sobol sequence), and selecting (20) points using Botlzmann sampling [19] of the 1024 initial points, according to their acquisition function utilities.

Combining PR with trust regions: When combining PR with the trust regions used in TuRBO we only use a trust region over the continuous parameters and discrete ordinals with at least 3 values. While methods like CASMOPOLITAN uses a Hamming distance for the trust regions over the categorical parameters, we choose not to do so as there is no natural way of efficiently optimizing PR using gradient-based methods. Finally, we do not use a trust region over the Boolean parameters as the trust region will quickly shrink to only include one possible value. We use the same hyperparameters as TuRBO [22] for unconstrained problems and SCBO [21] in the presence of outcome constraints, including default trust region update settings.

Casmopolitan: We use the official implementation of CASMOPOLITAN—which is available at `https://github.com/xingchenwan/Casmopolitan` under the MIT licence—but modify it where appropriate to additionally handle the ordinal variables. Specifically, the ordinal variables are treated as continuous when computing the kernel. However, during interleaved search, ordinal variables are searched via local search similar to the categorical variables. We use a set of CASMOPOLITAN hyperparameters (i.e. success/failure sensitivity, initial trust region sizes and expansion factor) recommended by the authors.

HyBO: We use the official implementation of HyBO at `https://github.com/aryandeshwal/HyBO`, which is licensed by the University of Amsterdam. We use the default hyperparameters recommended by the authors in all the experiments, and we use the full HyBO method with marginalization treatment of the hyperparameters as it has been shown to perform stronger empirically [14].

C.3 Gaussian process regression

When there are no categorical variables we use k_{ordinal} which is a product of an isotropic Matern-5/2 kernel for the binary parameters and a Matern-5/2 kernel with ARD for the remaining ordinal parameters. In the presence of categorical parameters, this kernel is combined with a categorical kernel [47] k_{cat} as $k_{\text{cat}} \times k_{\text{ordinal}} + k_{\text{cat}} + k_{\text{ordinal}}$. We use a constant mean function. The GP hyperparameters are fitted using L-BFGS-B by optimizing the log-marginal likelihood. The ranges for the ordinal parameters are rescaled to $[0, 1]$ and the outcomes are standardized before fitting the GP.

C.4 Sample Average Approximation

As discussed in Section 4.3, we instead of directly optimizing the stochastic MC estimators, we instead use sample average approximation. The idea is to use common random numbers (i.e. a fixed set of samples) to reduce variance when comparing a stochastic function at different inputs by using the same random numbers. This leads to biased deterministic estimators that are lower-variance than their stochastic counterparts where random numbers are re-sampled at each step. Such techniques have been used in the context of BO in settings such as efficiently optimizing MC acquisition functions [1] and for optimizing risk measures of acquisition functions under random inputs [10].

Sampling a fixed set of points $\tilde{\boldsymbol{z}}_1, ..., \tilde{\boldsymbol{z}}_N \sim p(\boldsymbol{Z}|\boldsymbol{\theta})$ would be a poor choice because $p(\boldsymbol{Z}|\boldsymbol{\theta})$ can vary widely during AF optimization as $\boldsymbol{\theta}$ changes. Therefore, instead sample from $p(\boldsymbol{Z}|\boldsymbol{\theta})$ using reparameterizations provided in Table 4. Specifically, we reparameterize $\boldsymbol{Z}$ as a deterministic function $h(\cdot, \cdot)$ that operates component-wise on $\boldsymbol{\theta}$ and the random variable $\boldsymbol{U} = (u^{(1)}, ..., u^{(d_z)}), u^{(i)} \sim \text{Uniform}(0, 1)$: $\boldsymbol{Z} = h(\boldsymbol{\theta}, \boldsymbol{U})$. That is, each random variable $Z^{(i)}$ has a corresponding independent base random variable $U^{(i)}$ such that $Z^{(i)} = h(\theta^{(i)}, U^{(i)})$. Using a fixed a set of base samples $\{\tilde{\boldsymbol{u}}_i\}_{i=1}^N$, the samples of $\boldsymbol{Z}$ can be be computed as $\boldsymbol{z}_i = h(\boldsymbol{\theta}, \tilde{\boldsymbol{u}}_i)$. We note that even with fixed base samples, the samples $\{\boldsymbol{z}_i\}_{i=1}^N$ depends on $\boldsymbol{\theta}$, and hence, by using common *base* uniform samples, we obtain a deterministic estimator where the values of the samples $\tilde{\boldsymbol{z}}_1, ..., \tilde{\boldsymbol{z}}_N$ can still vary with $\boldsymbol{\theta}$.

Table 4: Discrete random variables and their reparameterizations in terms of a Uniform random variable $U \sim \text{Uniform}(0, 1)$ and θ via a deterministic function $h(\cdot, \cdot)$.

TYPE	RANDOM VARIABLE	REPARAMETERIZATION ($Z = h(\theta, U)$)
BINARY	$Z \sim \text{BERNOULLI}(\theta)$	$h(\theta, U) = \mathbb{1}(U < \theta)$
ORDINAL	$Z = \lfloor \theta \rfloor + B$, $B \sim \text{BERNOULLI}(\theta - \lfloor \theta \rfloor)$	$h(\theta, U) = \lfloor \theta \rfloor + \mathbb{1}(U < \theta - \lfloor \theta \rfloor)$
CATEGORICAL	$Z \sim \text{CATEGORICAL}(\boldsymbol{\theta})$	$h(\theta, U) = \min(\arg\max_{i=0}^{C-1} \mathbb{1}(U < \sum_c^i \theta^{(c)}))$

D Constrained and Multi-Objective Bayesian Optimization

In many practical problems, the black-box objective must be maximized subject to $V > 0$ black-box outcome constraints $f_c^{(v)}(\boldsymbol{x}, \boldsymbol{z}) \geq 0$ for $v = 1, ..., V$. See Gardner et al. [24] for a more in depth review of black-box optimization with black-box constraints and BO techniques for this class of problems.

In the multi-objective setting, the goal is to maximize (without loss of generality) a set of M objectives $f^{(1)}, ..., f^{(M)}$. Typically there is no single best solution, and hence the goal is to learn the Pareto frontier (i.e. the set of optimal trade-offs between objectives). In the multi-objective setting, the hypervolume indicator is a common metric for evaluating the quality of a Pareto frontier. See [20] for a review of multi-objective optimization.

E Wall times

In this section, we reported the AF optimization wall times for the experiments in the main text.

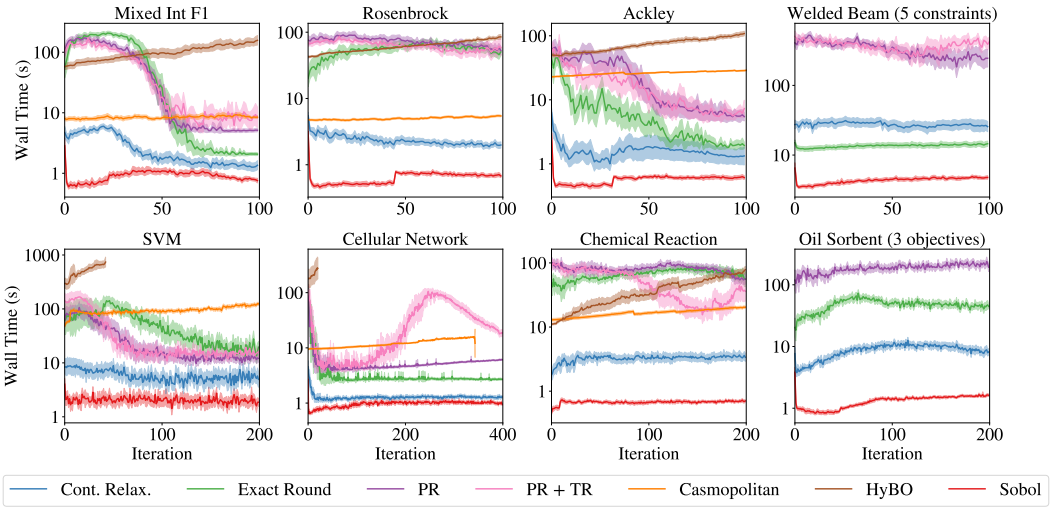

Figure 3: Wall time for candidate generation at each BO iteration in seconds. CONT. RELAX., EXACT ROUND, PR, and PR + TR are run on a single Tesla V100-SXM2-16GB GPU and other methods are run on an Intel Xeon Gold 6252N CPU.

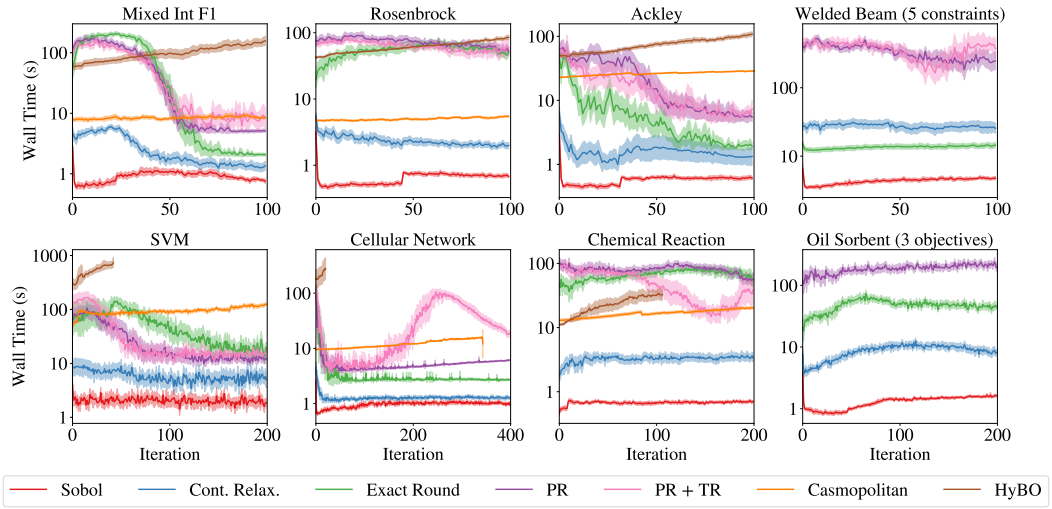

Figure 4: Wall time for candidate generation at each BO iteration in seconds. CONT. RELAX., EXACT ROUND, PR, and PR + TR are run on a single Tesla V100-SXM2-16GB GPU and other methods are run on an Intel Xeon Gold 6252N CPU.

F Probabilistic Reparameterization Sensitivity Analyses

F.1 Monte Carlo samples

The main text considers 1024 MC for PR. We consider 128, 256, and 512 samples, in addition to the default of 1024. For problems with discrete spaces that are enumerable, we also consider analytic PR. We do not find statistically significant differences between the final regret of any of these configurations (Figure 5). Run time is linear with respect to MC samples, and so substantial compute savings are possible when fewer MC samples are used (Figure 6). We observe comparable performance between PR with 1,024 MC samples and as few as 128 MC samples. With 64 or fewer MC samples, we observe performance degradation with respect to log regret in Figure 7, although wall time is considerably faster for fewer 64 or less MC samples as shown in Figure 8.

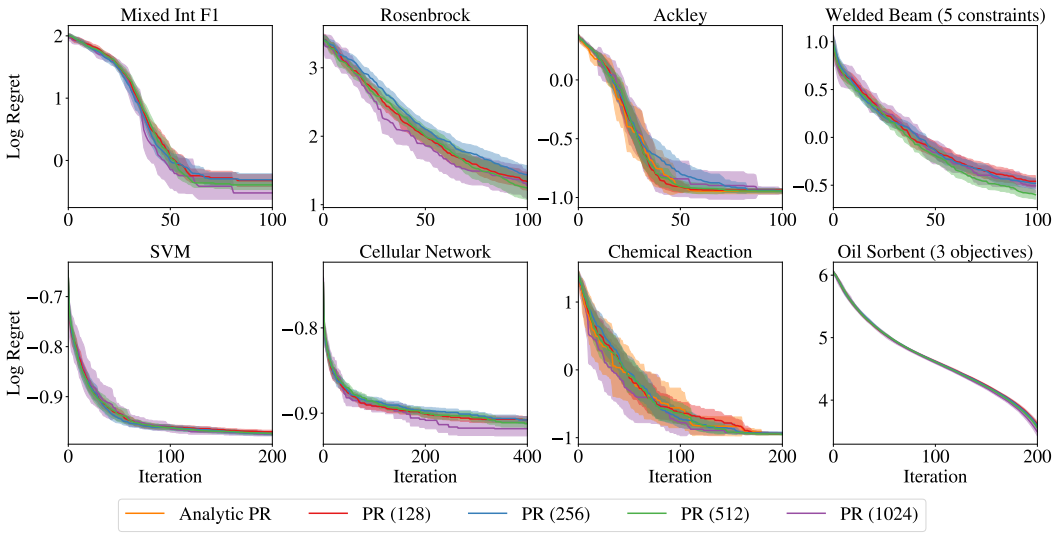

Figure 5: A sensitivity analysis of the optimization performance of PR with respect to the number of MC samples. We find that PR is robust to the number of MC samples, and the MC PR matches the performance of the analytic PR.

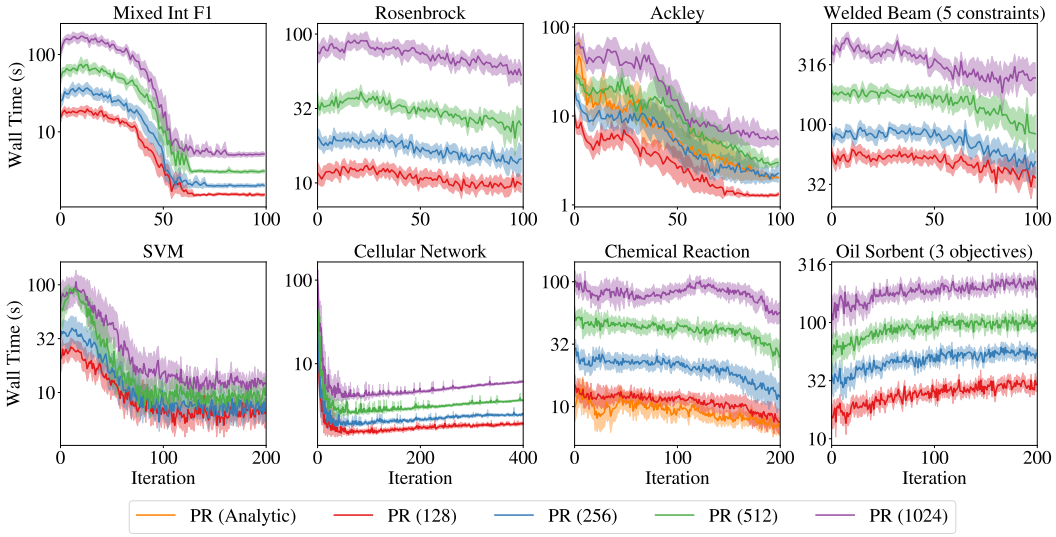

Figure 6: A sensitivity analysis of the wall time of PR with respect to the number of MC samples. We observe that wall time scales linearly with the number of MC samples, which is expected since we compute PR in $\frac{N}{32}$ chunks to avoid overflowing GPU memory.

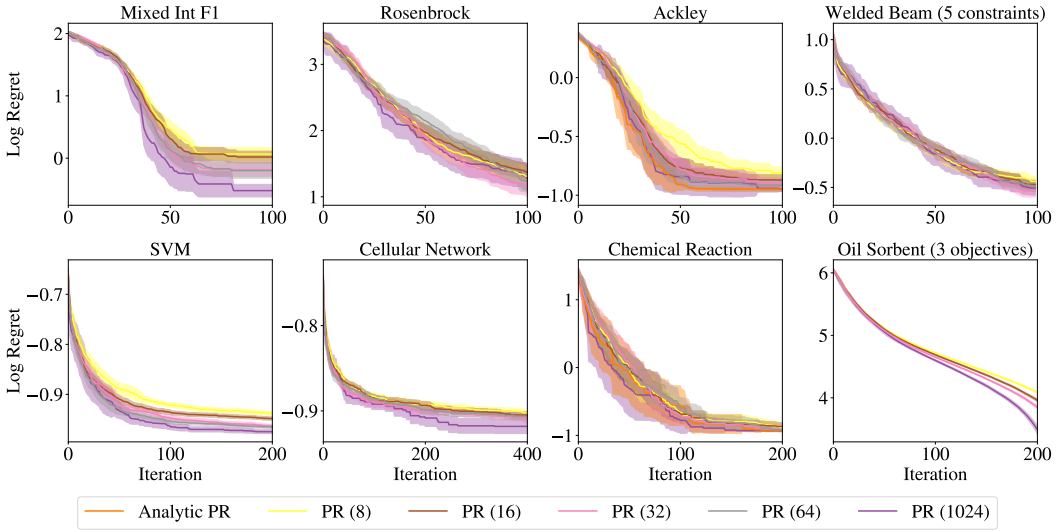

Figure 7: A sensitivity analysis of the optimization performance of PR with respect to a small number of MC samples (with samples between 8 and 64). Performance degrades slightly when few samples are used.

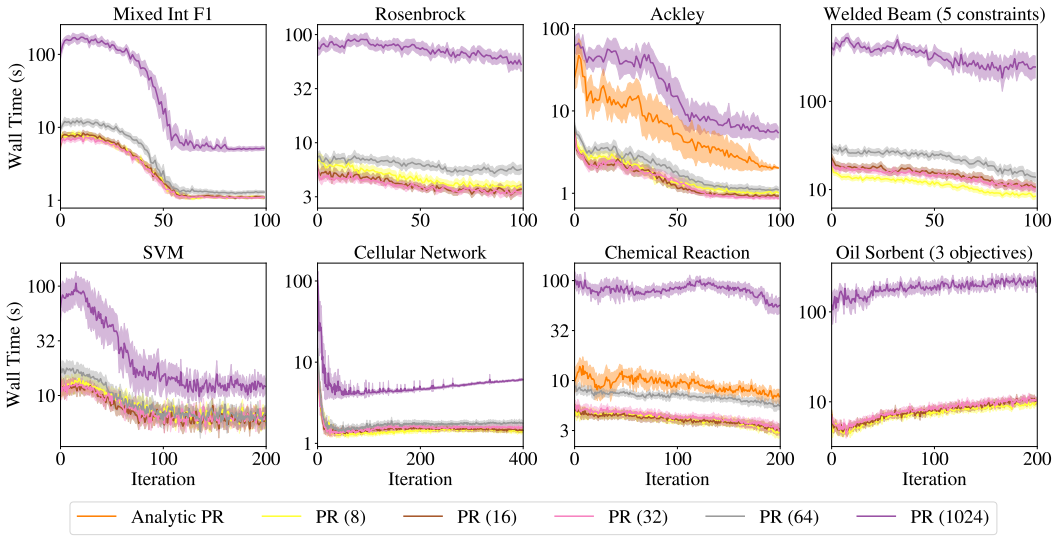

Figure 8: A sensitivity analysis of the wall time of PR with respect to the number of MC samples (with samples between 8 and 64). We observe that wall time scales linearly with the number of MC samples, which is expected since we compute PR in $\frac{N}{32}$ chunks to avoid overflowing GPU memory.

Since PR performs well with 128 MC samples, we report the main results from the paper using 128 samples with PR and PR + TR in Figures 9 and 10. We observe comparable performance to PR and PR + TR with 1,024 MC samples and much faster wall times—which are competitive with other methods.

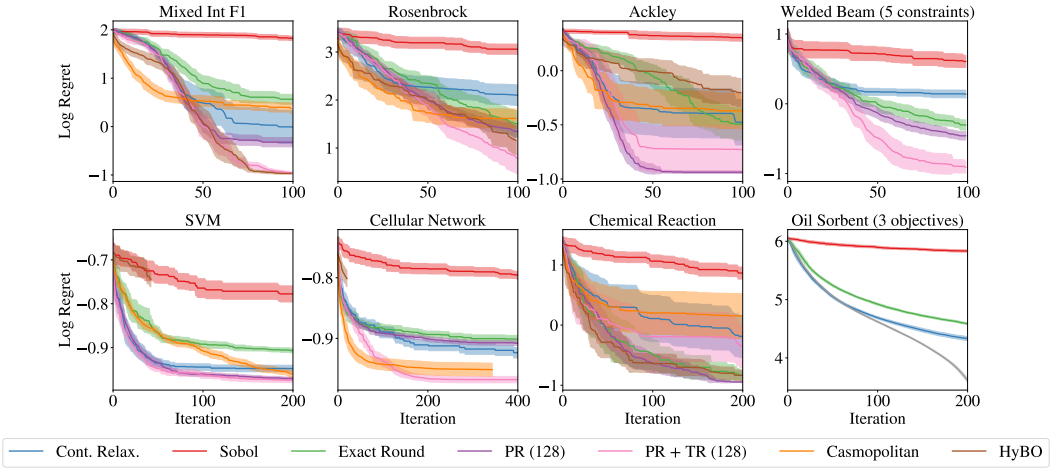

Figure 9: PR (or PR + TR) using 128 MC samples consistently outperforms alternatives with respect to log regret.

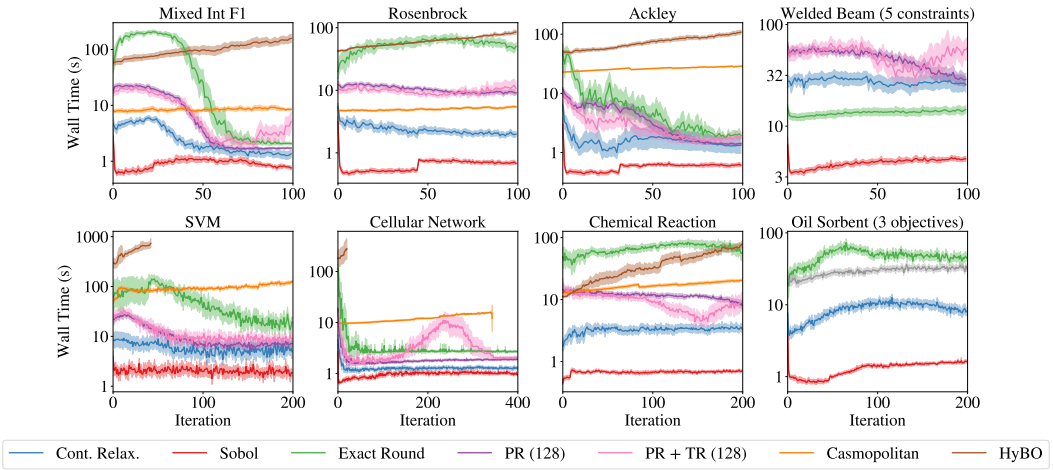

Figure 10: Wall time for candidate generation at each BO iteration in seconds with PR and PR + TR using 128 MC samples. CONT. RELAX., EXACT ROUND, PR, and PR + TR are run on a single Tesla V100-SXM2-16GB GPU and other methods are run on an Intel Xeon Gold 6252N CPU.

F.2 Effect of τ in Transformation

Throughout the main text, we use $\tau = 0.1$, which we selected based on the observation that it provides a reasonable balance between retaining non-zero gradients of $g(\phi)$ with respect to ϕ and allowing θ to become close to 0 or 1. This observation is obvious from Figure 11.

We further examine the effects of this hyperparameter by testing values of τ across four orders of magnitude from 10^0 to 10^{-3} in the following sensitivity analysis. We find that our default choice of 0.1 tends to perform well across all benchmark problems considered in this work.

As $\tau \to 0$, the θ can take more extreme values, but the gradient of the transformation with respect to ϕ also moves closer to zero. For larger values of τ, the gradient of the transformation with respect to ϕ is larger, but θ has a more limited domain with less extreme values. We find that $\tau = 0.1$ is a robust setting across all experiments.

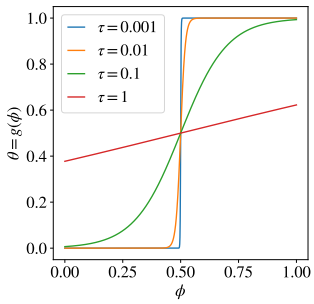

Figure 11: A comparison of the reparameterization of θ under various choices of τ. We observe that $\tau = 0.1$ provides a reasonable balance between retaining non-zero gradients of $g(\phi)$ with respect to ϕ and allowing θ to become close to 0 or 1.

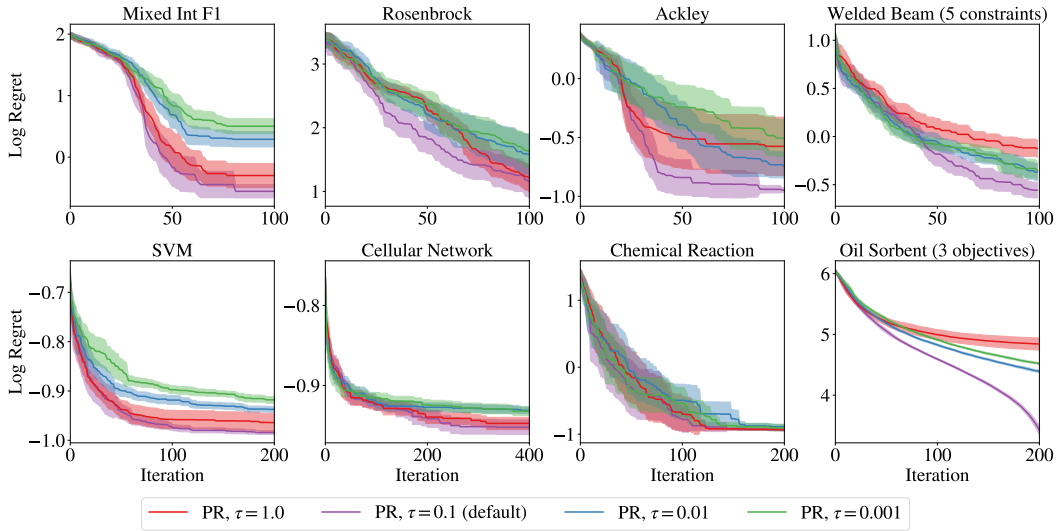

Figure 12: A sensitivity analysis of the log regret of PR with respect to the τ. We find $\tau = 0.1$ to be quite robust across experiments.

G Alternative methods

G.1 Straight-through gradient estimators

An alternative approach to using approximating the gradients under exact rounding using finite differences is to approximate the gradients using straight-through gradient estimation (STE) [3]. The idea of STE is to approximate the gradient of a function with the identity function. In our setting, the gradient of the discretization function with respect to its input is estimated using an identity function. Using this estimator enables gradient-based AF optimization, even though the true gradient of the discretization function is zero everywhere that it is defined. Although STEs have been shown to work well empirically, these estimators are not well-grounded theoretically. Their robustness and potential pitfalls in the context of AF optimization have not been well studied. Below, we evaluate the aforementioned EXACT ROUND + STE approach and show that it offers competitive optimization performance (Figure 13) with fast wall times (Figure 14), but does not quite match the optimization performance of PR on several benchmark problems.

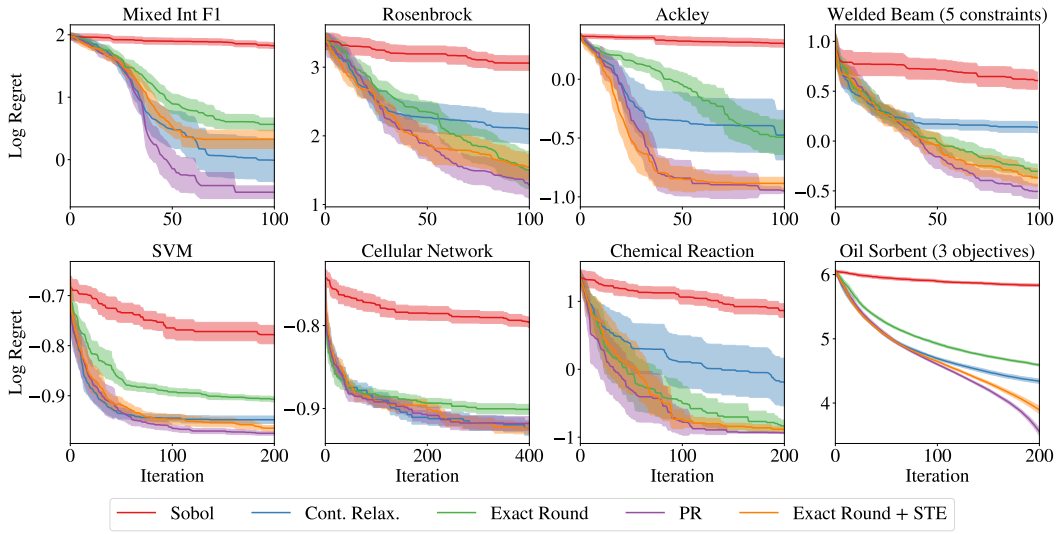

Figure 13: A comparison of exact rounding with straight-through gradient estimators versus other acquisition optimization strategies. Log regret on each problem. We report log hypervolume regret for Oil Sorbent and report the log regret of the best feasible objective for Welded beam.

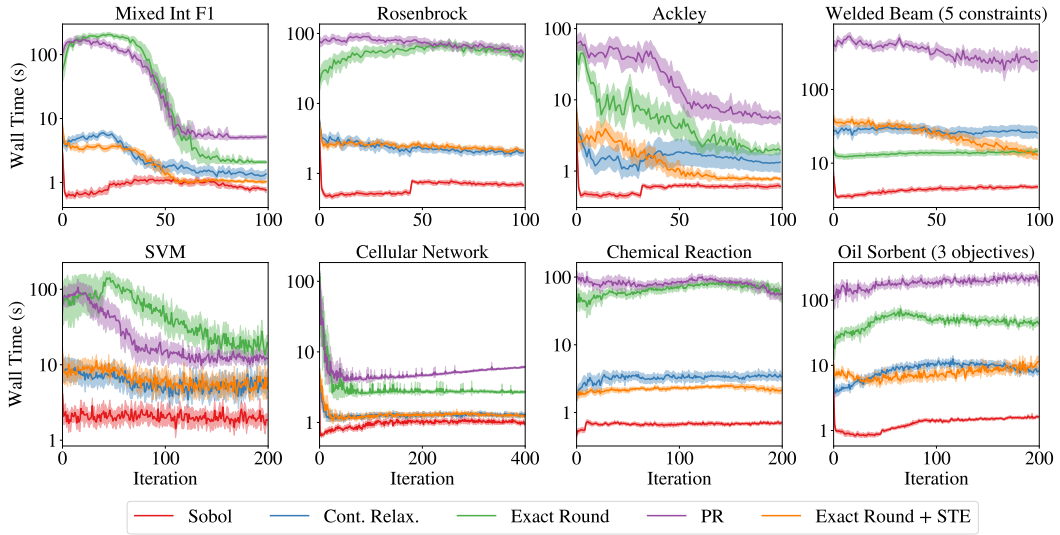

Figure 14: A comparison of wall times of exact rounding with straight-through gradient estimators versus other acquisition optimization strategies.

G.2 TR methods with alternative optimizers

In this section, we consider alternative methods to PR for optimizing AFs using within trust regions. The results in Figure 15 show that PR is a consistent best optimizer using TRs, but that STEs work quite well with TRs in many scenarios.

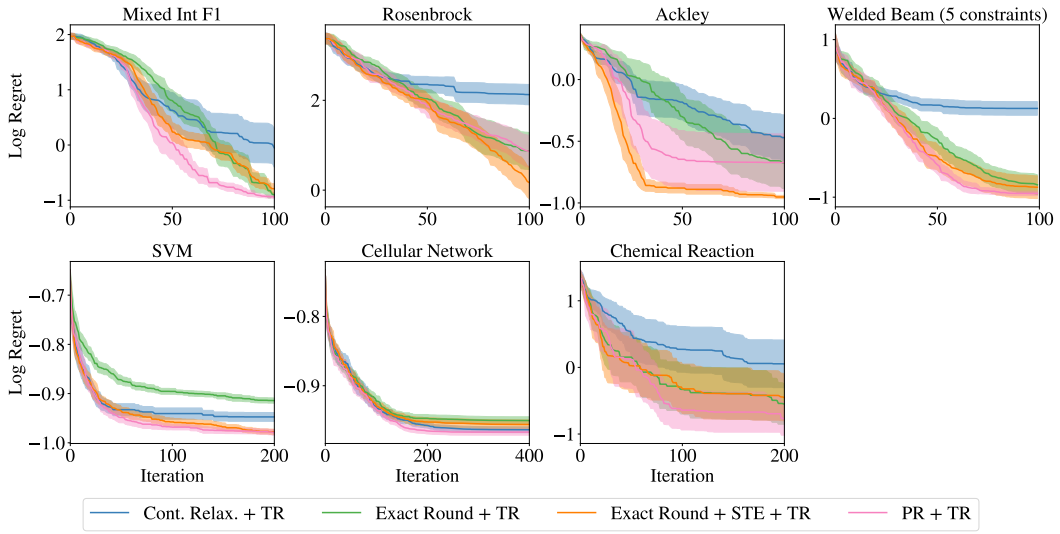

Figure 15: A comparison of TR methdods with different acquisition optimization strategies. Log regret on each problem. We report log hypervolume regret for Oil Sorbent and report the log regret of the best feasible objective for Welded beam.

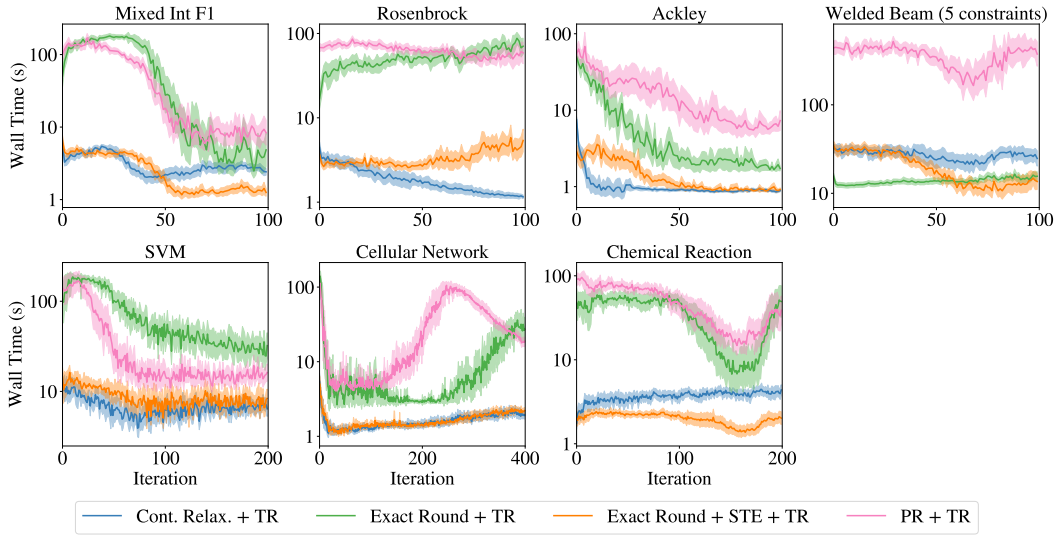

Figure 16: A comparison of wall times of TR methdods with different acquisition optimization strategies.

H Alternative categorical kernels

In this section, we demonstrate that PR can be used with arbitrary kernels over the categorical parameters including those that require discrete inputs (which CONT. RELAX. is incompatible with). Specifically for the categorical parameters, we compare using a Matérn-5/2 Kernel with one-hot encoded categoricals, a Categorical kernel (default), and a latent embedding kernel [65]. For the latent embedding kernel, we follow Pelamatti et al. [43] and use a 1-d latent embedding for categorical parameters where the cardinality is less than or equal to 3 and a 2-d embedding for categorical parameters where the cardinality is greater than 3. For each latent embedding, we use an isotropic Matérn-5/2 kernel and use product kernel across the kernels for the categorical, binary, ordinal, and continuous parameters. The results presented in Figure 18 show that latent embedding and categorical kernels have much lower variance across replications on the chemical reaction problem.

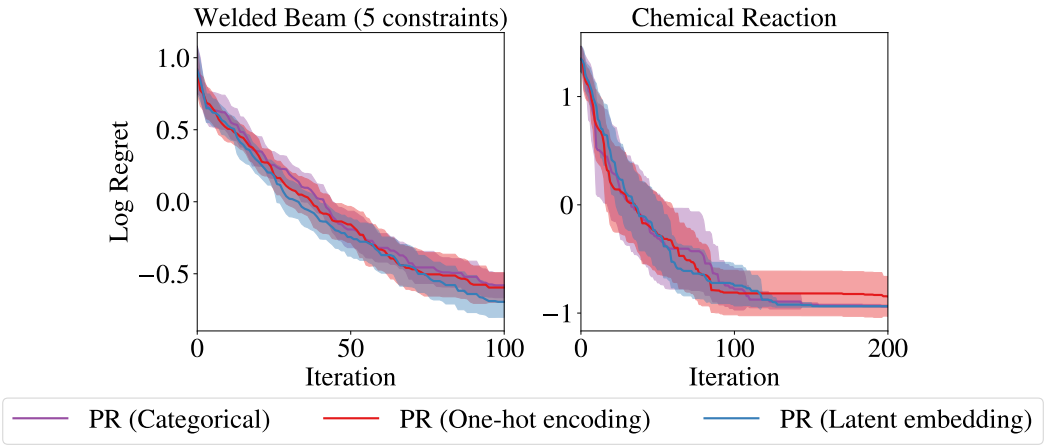

Figure 17: A comparison of different kernels over categorical parameters. Left: Welded beam has one categorical parameter, metal type (4 levels). Right: Chemical reaction has three categorical parameters, solvent, base, and ligand (with 4, 12, and 4 levels, respectively).

I Alternative Acquisition Functions

In this section, we compare PR with expected improvement (EI) against PR with upper confidence bound (UCB). For UCB, we set the hyperparameter β in each iteration using the method in Kandasamy et al. [33]. Although UCB comes enjoys bounded regret [51], we find empirically that EI works better on most problems.

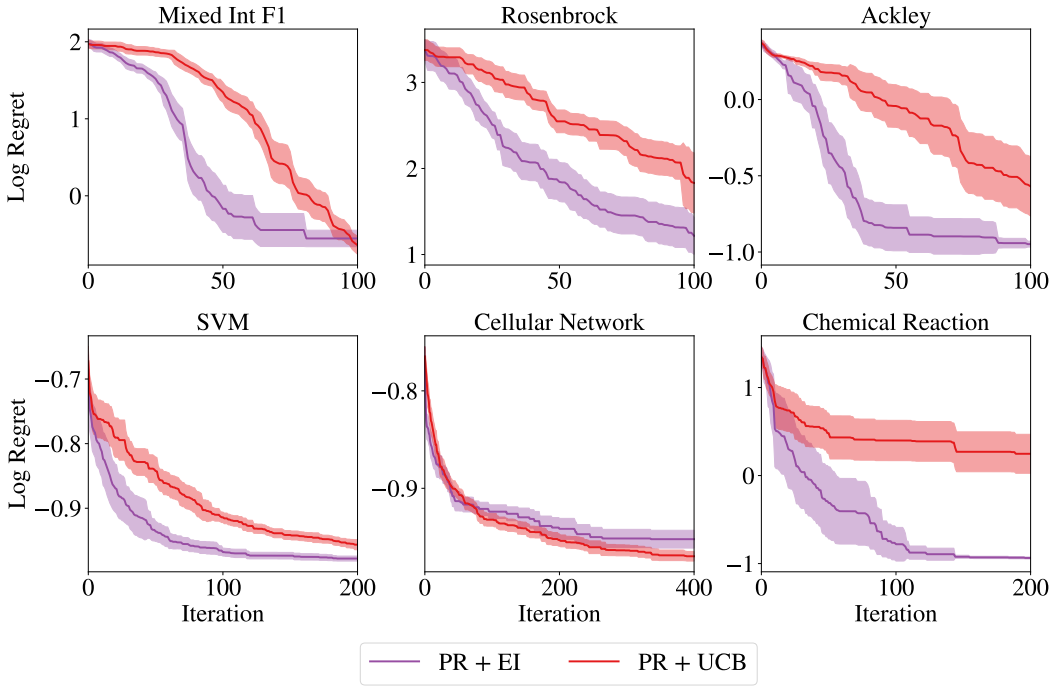

Figure 18: A comparison of expected improvement (EI) and upper confidence bound (UCB) acquisition functions with PR.

J Additional Results on Optimizing Acquisition Functions

In this section, we provide additional results on various approaches for optimizing acquisition functions using the same evaluation procedure as in the main text. We use 50 replications.

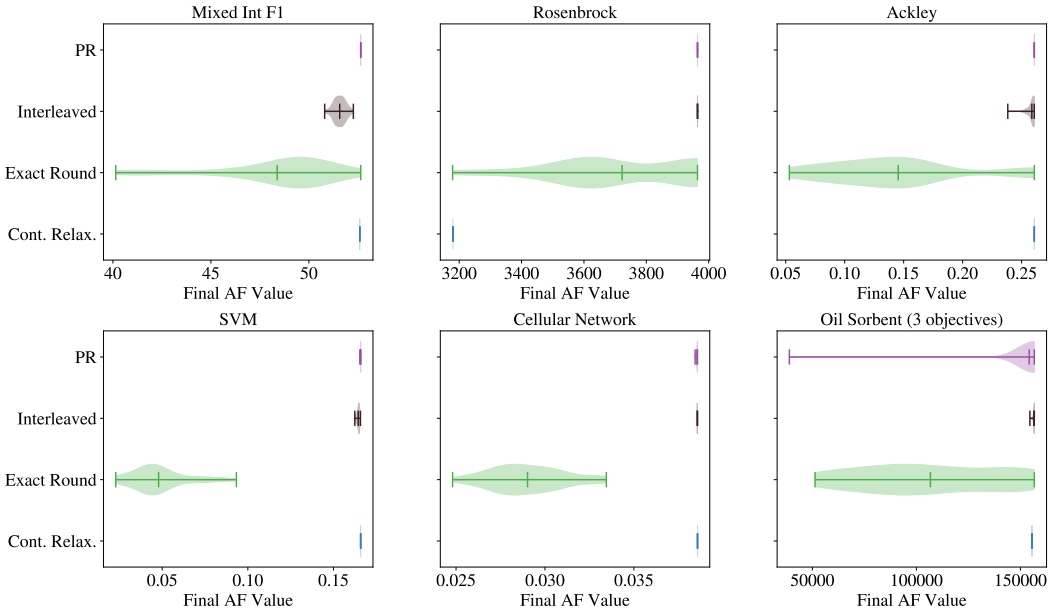

Figure 19: A comparison of methods for optimizing acquisition functions.

K Stochastic vs Deterministic Optimization

In this section, we compare optimizing PR with stochastic and deterministic optimization methods. For stochastic methods, we use stochastic gradient ascent (SGA) with various initial learning rates between 10^{-1} and 10^{-3}. The learning rate is multiplied by 0.1 every 30 steps following the approach in Huang et al. [29]. For SGA, the MC estimators of PR and its gradient resample the base samples at each evaluation. For deterministic optimization, base samples are kept fixed. SGA is run for 200 iterations and L-BFGS-B is run for a maximum of 200 iterations. In Figure 20, we observe that SGA works well on some problems with some learning rates, but SGA is sensitive to the choice of learning rate. Furthermore, we observe that the best learning rate varies from problem to problem. Deterministic optimization consistently performs better than stochastic optimization and has lower variance across replications.

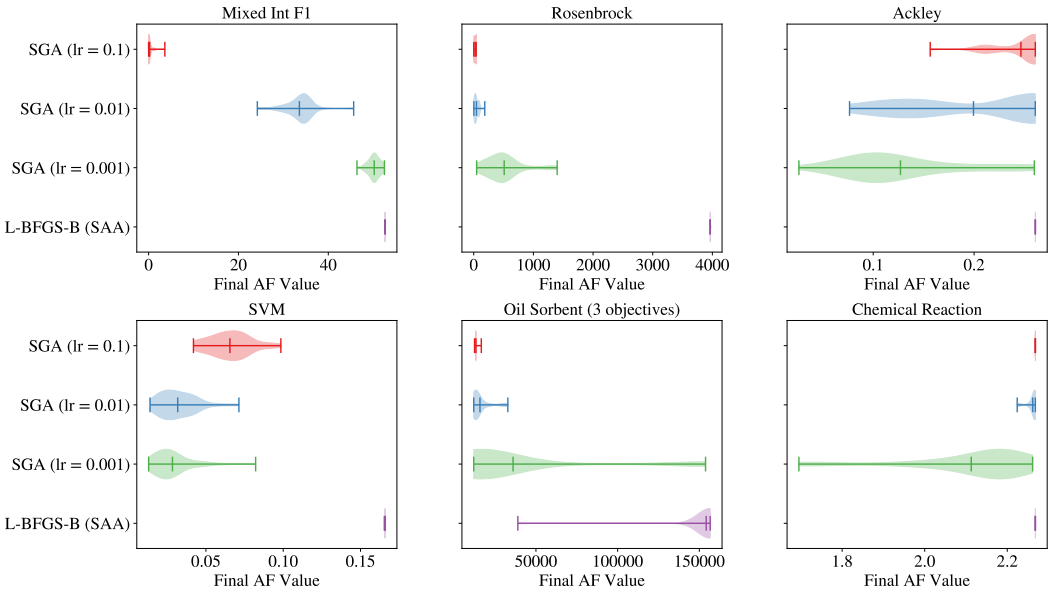

Figure 20: A comparison of PR using stochastic and deterministic optimization methods. The initial learning rate for stochastic gradient ascent is given in parentheses.

L Discussion of related work

Although the reparameterization trick used by Wilson et al. [61] is in a similar vein to PR, Wilson et al. [61] reparameterizes an existing multivariate normal random variable in terms of standard normal random variables and then proposes to use sample-path gradient estimators. In contrast, our approach introduces a new probabilistic formulation using discrete probability distributions and uses likelihood-ratio-based gradient estimators since sample-path gradients cannot be computed through discrete sampling.

M Comparison with an Evolutionary Algorithm

In Figures 21 and 21, we compare against the evolutionary algorithm PortfolioDiscreteOnePlusOne, which is the recommended algorithm for discrete and mixed search spaces in the Nevergrad package [44]. We find that PR significant outperforms this baseline with respect to log regret, but is slower than the evolutionary algorithm with respect to wall time.

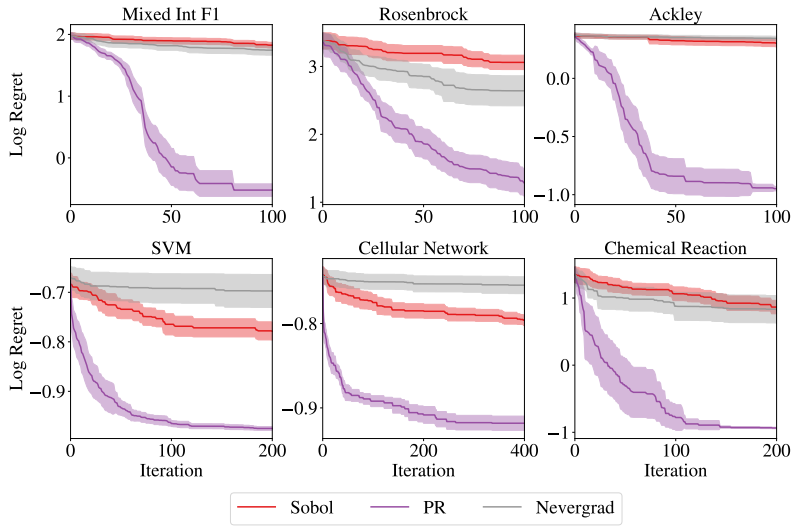

Figure 21: A comparison with an evolutionary algorithm with respect to log regret.

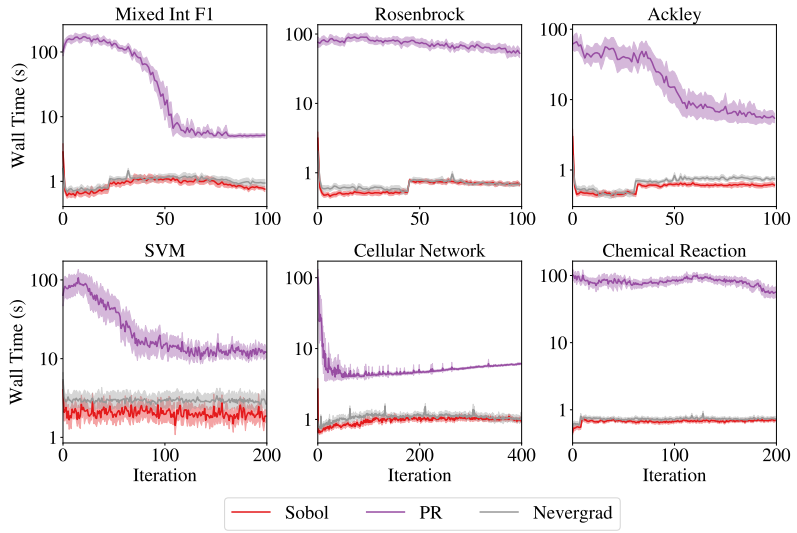

Figure 22: A comparison with an evolutionary algorithm with respect to wall time.

N Comparison with Enumeration

When computationally feasible, the gold standard for acquisition optimization over discrete and mixed search spaces is to enumerate the discrete options and optimize any continuous parameters for each discrete configuration (or simply evaluated each discrete configuration for fully discrete spaces). In Figures 23 and 24 we compare PR (with 1024 MC samples) against enumeration and show that PR achieves log regret performance that is comparable to the gold standard of enumeration and does so in less wall time.

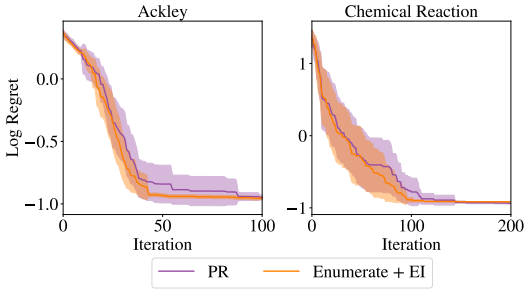

Figure 23: A comparison with an enumeration (gold standard) with respect to log regret.

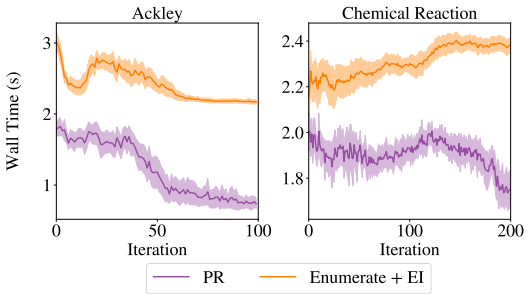

Figure 24: A comparison with enumerationwith respect to wall time.

O Acquisition Function Optimization at a Given Wall Time Budget

In Figure 25, we provide additional starting points (64 points, rather than 20) to other non-PR methods in order to provide them with additional wall-time budget. We find that using PR with 64 MC samples, PR provides rapid convergence compared to other baselines and therefore is a good optimization routine for any wall time budget.

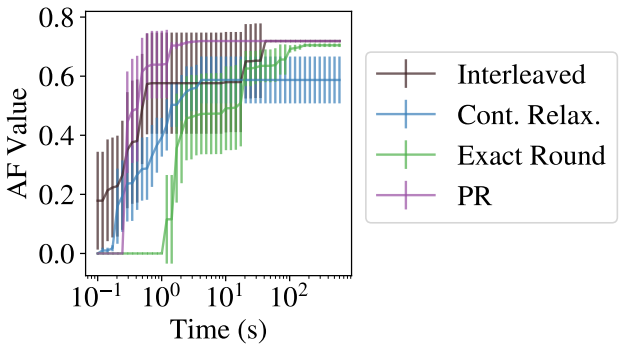

Figure 25: A comparison of methods for optimizing acquisition functions at a given wall time budget.