Bayesian Optimization in Adverse Scenarios

Samuel James Daulton

St. Catherine's College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hilary 2023

Abstract

Optimization problems with expensive-to-evaluate objective functions are ubiquitous in scientific and industrial settings. Bayesian optimization has gained widespread acclaim for optimizing expensive (and often black box) functions due to its theoretical performance guarantees and empirical sample efficiency in a variety of settings. Nevertheless, many practical scenarios remain where prevailing Bayesian optimization techniques fall short. We consider four such scenarios. First, we formalize the optimization problem where the goal is to identify robust designs with respect to multiple objective functions that are subject to input noise. Such robust design problems frequently arise, for example, in manufacturing settings where fabrication can only be performed with limited precision. We propose a method that identifies a set of optimal robust designs, where each design provides probabilistic guarantees jointly on multiple objectives. Second, we consider sample-efficient high-dimensional multi-objective optimization. This line of research is motivated by the challenging task of designing optical displays for augmented reality to optimize visual quality and efficiency, where the designs are specified by high-dimensional parameterizations governing complex geometries. Our proposed trust-region based algorithm yields order-of-magnitude improvements in sample complexity on this problem. Third, we consider multi-objective optimization of expensive functions with variable-cost, decoupled, and/or multi-fidelity evaluations and propose a Bayes-optimal, non-myopic acquisition function, which significantly improves sample efficiency in scenarios with incomplete information. We apply this to hardware-aware neural architecture search where the objective, on-device latency and model accuracy, can often be evaluated independently. Fourth, we consider the setting where the search space consists of discrete (and potentially continuous) parameters. We propose a theoretically grounded technique that uses a probabilistic reparameterization to transform the discrete or mixed inner optimization problem into a continuous one leading to more effective Bayesian optimization policies. Together, this thesis provides a playbook for Bayesian optimization in several practical adverse scenarios.

Bayesian Optimization in Adverse Scenarios

Samuel James Daulton

St. Catherine's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2023

This thesis is dedicated to
Sandra Lynn Daulton
for her unwavering affirmation
in all pursuits of education.

Acknowledgements

Personal

Thank you to my advisor Mike Osborne for your support throughout my DPhil. I would also like to thank Max Balandat and Eytan Bakshy for your advice, mentorship and support over the last 5 years.

Thanks to my parents Sandra Daulton and James Daulton; my step-parents Roz Ho and Michael Zeidman; my sister Melanie Daulton; and all of my family and friends.

And thanks to my partner Sara Violett for tolerating late nights and Sunday workdays—at the expense of adventures and contributing to good housekeeping.

Institutional

I am grateful for the support of Meta and the University of Oxford during my DPhil.

Abstract

Optimization problems with expensive-to-evaluate objective functions are ubiquitous in scientific and industrial settings. Bayesian optimization has gained widespread acclaim for optimizing expensive (and often black box) functions due to its theoretical performance guarantees and empirical sample efficiency in a variety of settings. Nevertheless, many practical scenarios remain where prevailing Bayesian optimization techniques fall short. We consider four such scenarios. First, we formalize the optimization problem where the goal is to identify robust designs with respect to multiple objective functions that are subject to input noise. Such robust design problems frequently arise, for example, in manufacturing settings where fabrication can only be performed with limited precision. We propose a method that identifies a set of optimal robust designs, where each design provides probabilistic guarantees jointly on multiple objectives. Second, we consider sample-efficient high-dimensional multi-objective optimization. This line of research is motivated by the challenging task of designing optical displays for augmented reality to optimize visual quality and efficiency, where the designs are specified by high-dimensional parameterizations governing complex geometries. Our proposed trust-region based algorithm yields order-of-magnitude improvements in sample complexity on this problem. Third, we consider multi-objective optimization of expensive functions with variable-cost, decoupled, and/or multi-fidelity evaluations and propose a Bayes-optimal, non-myopic acquisition function, which significantly improves sample efficiency in scenarios with incomplete information. We apply this to hardware-aware neural architecture search where the objective, on-device latency and model accuracy, can often be evaluated independently. Fourth, we consider the setting where the search space consists of discrete (and potentially continuous) parameters. We propose a theoretically grounded technique that uses a probabilistic reparameterization to transform the discrete or mixed inner optimization problem into a continuous one leading to more effective Bayesian optimization policies. Together, this thesis provides a playbook for Bayesian optimization in several practical adverse scenarios.

Contents

List of Figures

List of Tables

List of Abbreviations

AF . . . . . Acquisition function.

BO . . . . . Bayesian optimization.

CD . . . . . Competitive decoupling.

D-HV-KG . . Decoupled hypervolume knowledge gradient.

EHVI . . . . . Expected hypervolume improvement.

EI . . . . . Expected improvement.

GP . . . . . Gaussian Process.

IEP . . . . . Inclusion-exclusion principle.

MOBO . . . . Multi-objective Bayesian optimization.

MOO . . . . . Multi-objective optimization.

HV . . . . . Hypervolume.

HVI . . . . . Hypervolume improvement.

HV-KG . . . . Hypervolume knowledge gradient.

MARS . . . . . MVaR approximation via random scalarizations.

MC . . . . . Monte Carlo.

MF . . . . . Multi-fidelity.

MF-HV-KG . Multi-fidelity hypervolume knowledge gradient.

MORBO . . . Multi-objective regionalized Bayesian optimization

MVaR . . . . . Multi-variate value-at-risk.

NCD . . . . . Non-competitive decoupling.

NEHVI . . . . Noisy Expected Hypervolume Improvement.

NEI . . . . . Noisy expected improvement.

PR . . . . . Probabilistic Reparameterization.

q**EHVI** . . . . . q-Expected hypervolume improvement.

q**HV-KG** . . . q-Hypervolume knowledge gradient.

qNEHVI . . . q-Noisy Expected hypervolume improvement.

qParEGO . . . qParEGO algorithm.

qNParEGO . . qNParEGO algorithm.

TR . . . . . Trust region.

TS . . . . . Thompson sampling.

UCB . . . . . Upper confidence bound.

VaR . . . . . Value-at-risk.

1 Introduction

Contents

1.1 Motivation

Sample efficient optimization is critical in many scientific and industrial applications. For example, in the vaccine production process, freeze drying is an important step for increasing the storage lifetime of the vaccine [Mortier et al., 2016, Xie and Schenkendorf, 2019]. The freeze drying process is sensitive to experimental conditions such as the shelf temperature and the chamber pressure. In this scenario, a scientist is likely interested in identifying an optimal setting of the shelf temperature and chamber pressure to maximize the quality of the vaccine after freeze drying. Evaluating this objective function may require performing a wet lab experiment. In this case, the objective function would likely be treated as

a black box. To find the optimal experimental conditions, the objective function can be queried at different experimental conditions (design parameters) in order to receive potentially noisy observations of the objective. This approach is called zeroth order optimization, which is the de facto class of algorithm in the absence of information about the gradient of the objective.

In the vaccine production example, evaluating the objective would be expensive in terms of money and time. The monetary cost would include the cost to access to the wet lab bench, the cost of the technician to conduct the experiment, and the cost of the vaccine sample that must be freeze dried and potentially catastrophically altered. Furthermore, evaluating the objective would be expensive in terms of the time to freeze dry the vaccine and the time to measure its quality. When the objective is expensive-to-evaluate, sample efficiency is paramount.

Bayesian optimization is a popular approach for sample efficient optimization due to its strong empirical performance and theoretical performance guarantees in many scenarios [Shahriari et al., 2016, Frazier, 2018, Garnett, 2023]. Its widespread adoption in many fields has led to applying Bayesian optimization to challenging problems. These difficult applied optimization problems have motivated research on extending Bayesian optimization to a variety of problem classes including multi-objective (e.g. Emmerich et al. [2006], Knowles [2006], Hernandez-Lobato et al. [2016], Daulton et al. [2020]), discrete and mixed (e.g. Baptista and Poloczek [2018], Oh et al. [2019], Wan et al. [2021]), high-dimensional (e.g. Wang et al. [2016b], Eriksson et al. [2019], Griffiths and Hernandez-Lobato [2020]), robust (e.g. Bogunovic et al. [2018], Fröhlich et al. [2020], Cakmak et al. [2020], and multi-fidelity (e.g. Poloczek et al. [2017], Takeno et al. [2020], Wu et al. [2020a]) optimization. However, this thesis presents several real-world *adverse scenarios* where existing Bayesian optimization algorithms fall short.

1.2 Contributions

1.2.1 Robust Multi-Objective Bayesian Optimization

First, we consider the multi-objective optimization setting where the tunable parameters are subject to input noise. In this setting, the decision maker will seek to identify parameterizations that are robust to input perturbations jointly across the objectives. For example, in the vaccine manufacturing example, there might be two objectives that the decision-maker seeks to maximize: the efficiency (minimize the time) of the freeze-drying step and vaccine quality. In addition, the setting experimental conditions may be subject to small amounts of error. Higher temperatures lead to more efficient freeze drying, but if the temperature is too high, catastrophic degradation of the vaccine can occur. In such a scenario, a decision maker may prefer a lower temperature to mitigate the risk of catastrophic loss. In the multi-objective setting, it is often important to be robust with respect to multiple objectives simultaneously. Through this lens, we formulate the multi-objective robust design problem in terms of identifying designs that provide probabilistic lower-bounds on the values of the objectives under input noise. We then propose Bayesian optimization algorithm for identifying designs with different optimal trade-offs between these lower bounds.

1.2.2 High-Dimensional Multi-Objective Bayesian Optimization

Second, we consider multi-objective optimization over high-dimensional search spaces. Although there has been significant research in Bayesian optimization methods for high-dimensional and multi-objective settings independently, there is a dearth of research at their intersection. The research in this chapter is motivated by a real world optimization problem where the goal is to optimize an optical display for augmented reality. The display is specified by over one hundred tunable parameters that define complex geometries. The goal is to identify designs that maximize the visual quality of the display image while simultaneously maximizing efficiency. Evaluating these objectives requires running computationally intensive

simulations, which are expensive in terms of time and money, or actually fabricating the display, which would be even more expensive and time consuming. However, evaluations can be highly parallelized by leveraging a computer cluster and the optimization budgets are often in the tens of thousands of evaluations. To tackle this problem, we propose an algorithm to address this class of high-throughput, large batch, multi-objective, high-dimensional optimization problems leveraging state-of-the-art techniques relying on performing Bayesian optimization in subspaces of the search domain.

1.2.3 Multi-Objective Bayesian Optimization with Partial Information

Third, we consider multi-objective Bayesian optimization with incomplete information. We consider two settings of incomplete information: (i) the setting where the objectives are *decoupled* and only a subset of the objectives (potentially with heterogenous evaluation costs) are observed for each design and (ii) the *multi-fidelity* setting where observations of the objectives at lower fidelities can be obtained at lower cost. Again, this work is motivated by the practical problem of multi-objective neural architecture search at Meta where the goal is to identify model architectures that are accurate and result in low on-device prediction latency [Eriksson et al., 2021]. In this case, accuracy is time-consuming to evaluate because the neural network must be trained, but many models can be trained in parallel on different compute nodes in a cluster. On the other hand, latency is much cheaper to evaluate and often does not require model training, but prediction time must be measured on specific devices such as mobile phones or specialized hardware. Often, very few devices are available. If the goal were to identify the optimal designs within a budget of end-to-end wall time, it would likely be advantageous to exploit all evaluation resources and suggest and dispatch designs to each objective asynchronously. To tackle this problem, we propose a non-myopic multi-objective Bayesian optimization technique that can suggest both designs and the manner in which those designs

should be evaluated (which objectives or at what fidelity level) in order to exploit incomplete information and improve optimization performance.

1.2.4 Discrete and Mixed Bayesian Optimization

Fourth, we consider the general problem of using Bayesian optimization over discrete and mixed search spaces. This line of research is applicable to many optimization problems including optimizing chemical reactions in order to maximize the yield [Shields et al., 2021]. In this scenario, the design parameters include both categorical parameters such as the choice of base, solvent, and ligand and numeric parameters such as temperature and concentration. Discrete parameters can pose challenges to standard Bayesian optimization methods that focus on optimizing continuous parameters. Our contribution is to reformulate the decision making problem (selecting a design to evaluate) from a discrete or mixed optimization problem into a continuous optimization problem via a probabilistic reparameterization. This approach is applicable across many problem settings, including for instance both the single objective and the multi-objective settings.

1.3 Thesis Outline

The thesis proceeds as follows:

- Chapter 2 provides background material on Bayesian Optimization.
- Chapter 3 formalizes the problem of robust multi-objective Bayesian optimization under input noise and present as a novel Bayesian optimization algorithm for this setting.
- Chapter 4 considers multi-objective Bayesian optimization over high-dimensional search spaces and proposes and applies a practical method for solving exceptionally challenging real world optimization problems.
- Chapter 5 proposes a new lookahead acquisition function for multi-objective Bayesian optimization with incomplete information (including multi-fidelity observations).

- Chapter 6 revisits Bayesian optimization over discrete and mixed search spaces and proposes a new technique for this setting.
- Chapter 7 offers a discussion of the contributions and a look toward to the future.

2
Background

Contents

2.1 Bayesian Optimization

Optimizing an expensive-to-evaluate function often requires sample efficiency: that is, the objective must be optimized by querying the objective at very few samples of input parameters $\boldsymbol{x}$ from the search space $\mathcal{X} \subset \mathbb{R}^D$. Typically, we are interesting performing optimization over a search space $\mathcal{X}$ that is a compact, hyperrectangular subset of $\mathbb{R}^D$. Without loss of generality, we consider optimization problems, where the goal is to maximize an objective function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ over $\mathcal{X}$: $\arg\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$. Although the proposed algorithms are applicable grey-box [Astudillo and Frazier, 2022] or white-box functions (where part or all, respectively, of the function has known behavior or a known expression), typically we consider the setting where f is a black-box function with no known analytic expression. Furthermore, we consider the setting where observations of the gradient of $f(\boldsymbol{x})$ with

respect to $\boldsymbol{x}$ cannot be obtained. In this setting, zeroth order optimization methods are typically employed to optimize the function f by querying f at different input locations $\boldsymbol{x}$ and receiving (potentially noisy) observations of $f(\boldsymbol{x})$. Typically the goal of an optimization algorithm is to minimize the simple regret $f(\boldsymbol{x}^*) - f(\hat{\boldsymbol{x}}^*)$, where $\boldsymbol{x}^*$ is a maximizer of f over $\mathcal{X}$ and $\hat{\boldsymbol{x}}^*$ is the algorithm's recommended best point. In the noiseless setting, often $\hat{\boldsymbol{x}}^*$ is chosen to be the best previously evaluated design $\hat{\boldsymbol{x}}^* = \arg\max_{\boldsymbol{x} \in \{\boldsymbol{x}\}_{i=1}^N} f(\boldsymbol{x})$, where N is the number of previously evaluated points. In contrast with cumulative regret, simple regret is not effected by the regret incurred to find the best point, but rather only the regret of the best point itself.

Many types of algorithms exist for black-box optimization including evolutionary strategies (e.g. Hansen [2007]), random algorithms (e.g. random search), and model-based approaches (e.g. Regis and Shoemaker [2013]). Bayesian optimization (BO) is a popular model-based sequential optimization approach due to its empirical sample-efficiency and theoretical performance guarantees [Shahriari et al., 2016, Frazier, 2018, Garnett, 2023]. BO relies on a Bayesian surrogate model of f to provide both point estimate predictions and uncertainty estimates. In order to select one or more new designs to evaluate, an acquisition function (AF) that leverages the surrogate model is used to quantify the value of evaluating a design (or batch of designs) on the objective function. Although evaluating the objective function itself is expensive, evaluating the surrogate model is usually relatively cheap and therefore numerical optimization (often gradient-based optimization) can be used to optimize the acquisition function to select one or more promising new designs. After selecting a new design $\boldsymbol{x}$, the objective function is queried and potentially noisy observation is received. Typically, the observations is assumed to be noiseless or the observations subject to zero-mean Gaussian homoskedastic noise[1] with variance σ^2 and the observation is given by $y = f(\boldsymbol{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Throughout this work, we use a Gaussian Process (GP) as the probabilistic surrogate model because GPs are highly flexible and capable of modeling a wide variety of functions, they have been observed to have well-calibrated uncertainty

[1]We also consider heteroskedastic noise in various parts of this thesis and in those cases observations of the noise level are received for each observation.

estimates, they lead to theoretical guarantees in the context of BO in a number of settings [Srinivas et al., 2010], and under Gaussian likelihoods, their predictive mean, predictive variance, and marginal likelihood (and their derivatives) can expressed analytically.

2.1.1 Acquisition Functions

A wide variety of acquisition functions have been proposed for Bayesian optimization. In this background section, we review some common improvement-based acquisition functions although many other alternatives exist including optimistic acquisition functions such as upper confidence bound (UCB) [Srinivas et al., 2010] and information theoretic acquisition functions [Hennig and Schuler, 2012, Hernández-Lobato et al., 2014, Wang and Jegelka, 2017, Ru et al., 2018].

Expected improvement (EI) [Jones et al., 1998] is commonly used acquisition function, which quantifies the expected improvement over the best incumbent value that will be obtained by evaluating a point $\boldsymbol{x}$ under the GP posterior:

$$\alpha_{\text{EI}}(\boldsymbol{x}) = \mathbb{E}[[f(\boldsymbol{x}) - f_N^*]_+],$$

where $[\cdot]_+$ denotes the $\max(\cdot, 0)$ operation and f_N^* is the best observed value. The EI of a single point can be computed analytically leveraging the GP's predictive distribution, but computing the EI of a batch of design points is not analytically tractable and typically Monte Carlo (MC) estimation is used [Wang et al., 2016a, Wilson et al., 2018]. The classical EI acquisition function assumes that the inputs are noise-free. In the presence of observation noise, the best incumbent value will often be unidentifiable, in which case it is advantageous to integrate over the best incumbent value [Letham et al., 2019].

When the simple regret performance metric is evaluated based on the best evaluated (in-sample) design, EI is one-step Bayes optimal by definition. A common alternative is to instead consider *inference regret*, where the best point can be chosen over the entire search space as the best point $\arg\max_{\boldsymbol{x}\in\mathcal{X}} \mu(\boldsymbol{x})$ under model's posterior mean function μ. Under this formulation, the one-step Bayes optimal

acquisition function is the Knowledge Gradient (KG) [Frazier et al., 2008], which quantifies the expected improvement in the maximizer of the posterior mean if we were to evaluate a new design $\boldsymbol{x}$:

$$\alpha_{\text{KG}}(\boldsymbol{x}) = \mathbb{E}\left[\max_{\boldsymbol{x}' \in \mathcal{X}} \mu_{n+1}(\boldsymbol{x}') - \mu_n^*\right],$$

where μ_n^* is the maximum of the current posterior mean function conditioned on the previously observed data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and μ_{n+1} is the posterior mean function after conditioning additionally on the new observation $(\boldsymbol{x}, y)$. The expectation is over the random variable y, which μ_{n+1} is conditioned upon. KG can be computed analytically for discrete search spaces, but is analytically intractable for continuous search spaces and is typically approximated via MC estimation [Wu et al., 2017].

2.2 Multi-Objective Bayesian Optimization

In multi-objective optimization (MOO), the goal is to optimize a vector-valued function $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^M$ over the search space $\mathcal{X}$, where $M > 1$. Typically, there is no single best solution that simultaneously maximizes all objectives.[2] Instead, the goal is to identify the set of optimal designs with respect to Pareto dominance. We denote the components of $\boldsymbol{f}(\boldsymbol{x})$ by $[f^{(1)}, ..., f^{(M)}]$. An objective vector $\boldsymbol{f}(\boldsymbol{x})$ *Pareto dominates* another point $\boldsymbol{f}(\boldsymbol{x}')$, which we denote by $\boldsymbol{f}(\boldsymbol{x}) \succ \boldsymbol{f}(\boldsymbol{x}')$, if for all $m = 1, ..., M, f^{(m)}(\boldsymbol{x}) \geq f^{(m)}(\boldsymbol{x}')$ and there exists $m' \in \{1, ..., M\}$ such that $f^{(m')}(\boldsymbol{x}) > f^{(m')}(\boldsymbol{x}')$. A design is *Pareto optimal* if it is not Pareto dominated by any other design. Hence, our goal is to identify the *Pareto set* $\mathcal{X}^* = \{\boldsymbol{x} \mid \nexists\, \boldsymbol{x}' \; s.t. \; \boldsymbol{f}(\boldsymbol{x}') \succ \boldsymbol{f}(\boldsymbol{x})\}$ of optimal designs and its image under $\boldsymbol{f}$, which is called the *Pareto frontier* (PF) and is defined by $\mathcal{P}^* = \{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X}^*\}$.

Many evaluation criteria exist for assessing the quality of a Pareto frontier. One of the most popular is the hypervolume (HV) indicator [Zitzler et al., 2007], which, given a Pareto frontier $\mathcal{P}$ and a reference point $r \in \mathbb{R}^M$, is the M-dimensional

[2] We assume maximization of all objectives without loss of generality.

Lebesgue measure of the space that is dominated by the Pareto frontier and bounded from below by the reference point.[3] More formally, let

$$Z = \{\boldsymbol{z} \in \mathbb{R}^M : \exists\ \boldsymbol{y} \in \mathcal{P}\ s.t.\ \boldsymbol{y} \succ \boldsymbol{z} \succ \boldsymbol{r}\}.$$

Then, $\mathrm{HV}(\mathcal{P}, \boldsymbol{r}) = \int_{\mathbb{R}^M} \mathbb{1}_Z(\boldsymbol{z}) d\boldsymbol{z}$, where $\mathbb{1}_Z(\boldsymbol{z})$ denotes characteristic function of Z. We note that there are many criterion for evaluating the quality of a Pareto frontier [Zitzler et al., 2008], but in this work, we focus on HV due to it widespread use and non-domination principle. That is, if for every $\boldsymbol{y}$ in one Pareto frontier $\mathcal{P}$ there exists a point $\boldsymbol{y}'$ in another another Pareto frontier $\mathcal{P}'$ such that $\boldsymbol{y}' \succeq \boldsymbol{y}$, then $\mathrm{HV}(\mathcal{P}') \geq \mathrm{HV}(\mathcal{P})$. If in addition, there exists at least one $\boldsymbol{y} \in \mathcal{P}$ for which there exists a point $\boldsymbol{y}' \in \mathcal{P}'$ such that $\boldsymbol{y}' \succ \boldsymbol{y}$, then $\mathrm{HV}(\mathcal{P}') > \mathrm{HV}(\mathcal{P})$.

Hence, a common way of evaluating the performance of a multi-objective optimization algorithm is via the hypervolume simple regret of its Pareto frontier approximation $\mathcal{P}$: $\mathrm{HV}(\mathcal{P}^*) - \mathrm{HV}(\mathcal{P})$. Similar to the single objective case, often $\mathcal{P}$ is taken to be the Pareto frontier over the previous evaluated designs, but it can also be the image of the Pareto set recommended by the model over the entire search space [Hernandez-Lobato et al., 2016, Suzuki et al., 2020, Tu et al., 2022]. Additionally, we note that the Pareto set can be a potentially infinite set and that the result from an optimization algorithm will typically be a finite approximation.

As in the single objective setting, there is a rich literature of evolutionary-based optimization methods (e.g. [Deb et al., 2002]), but Bayesian optimization is renowned for its sample efficiency multi-objective optimization problems.

2.2.1 Acquisition Functions

Many acquisition functions have been proposed in the multi-objective Bayesian optimization setting. In this section, we review fundamentals for scalarization and hypervolume-based methods, but acknowledge that there are many alternative

[3]Typically, it is assumed that the reference point is supplied by the decision maker using domain knowledge [Yang et al., 2019], but it can also be set heuristically [Ishibuchi et al., 2011, Ponweiser et al., 2008]. Henceforth, we omit $\boldsymbol{r}$ as an argument to HV for brevity.

approaches such as information theoretic methods [Hernandez-Lobato et al., 2016, Suzuki et al., 2020, Tu et al., 2022].

Scalarization approaches are popular due to their simplicity and empirical performance. The idea is the transform the objectives into a single scalar objective via a weighted scalarization and then leverage an acquisition function for single objective optimization. Sampling scalarization weights when selecting new candidate design allows the BO algorithm to explore different trade-offs. However, not all scalarizing functions can recover the entire Pareto frontier. For example a linear scalarization can not identify solutions in concave regions of the Pareto frontier [Chugh, 2020]. Due to its ability to recover non-convex regions of the Pareto frontier, the Chebyshev scalarization has received significant attention [Knowles, 2006]: $s(\boldsymbol{y}) = \min_m(w^{(m)}y^{(m)})$, where $\boldsymbol{y}$ is an objective vector $w^{(m)}$ is the weight applied to the m^{th} objective and $y^{(m)}$ is the value of the m^{th} objective. A popular scalarization-based algorithm is ParEGO [Knowles, 2006], which models a random Chebyshev scalarization augmented with a linear term (to ensure non-dominated points are preferred) with GP and uses the EI acquisition function. Subsequent works have modeled the objectives directly (rather than the scalarized value), leveraged MC acquisition functions, and used composite objectives [Astudillo and Frazier, 2019] to extend ParEGO to batch, constrained, and noisy optimization [Daulton et al., 2020, 2021]. Random Chebyshev scalarizations has also been used with alternative acquisition functions [Paria et al., 2020] to obtain Bayes regret bounds in some scenarios, and more recent work has obtained hypervolume regret bounds with an alternative scalarization [Golovin and Zhang, 2020].

Given that the quality of a Pareto frontier is often evaluated using the hypervolume indicator, a natural approach is to use an acquisition function that is one-step Bayes optimal for maximizing the evaluation criterion. Furthermore, it has been observed that maximizing hypervolume leads to high quality Pareto frontier approximations with good coverage [Zitzler et al., 2003, Couckuyt et al., 2014, Yang et al., 2019]. Hence, the expected hypervolume improvement (EHVI) acquisition

function is often used. The expected hypervolume improvement given the current Pareto frontier $\mathcal{P}$ over the previously evaluated designs is given by

$$\alpha_{\text{EHVI}}(\boldsymbol{x}) = \mathbb{E}[\text{HV}(\mathcal{P} \cup \{\boldsymbol{f}(\boldsymbol{x})\}) - \text{HV}(\mathcal{P})].$$

EHVI can be computed analytically when objectives are modeled independently, and EHVI is differentiable with respect to $\boldsymbol{x}$ [Yang et al., 2019, Daulton et al., 2020]. In the parallel case, a closed form expression for qEHVI is not known and qEHVI is approximated by MC approximation [Daulton et al., 2020]. Although Daulton et al. [2020] show how to efficiently optimize qEHVI, the time complexity of evaluating qEHVI scales exponentially in the batch size q (although it is constant given infinite computing cores) [Daulton et al., 2020]. To address this, Daulton et al. [2021] propose a more scalable caching-based sequential greedy approach called qNEHVI that reduces the complexity to polynomial. Furthermore, qNEHVI is a principled approach in the noisy setting where the incumbent Pareto frontier is unknown. qNEHVI integrates over the possible function values at the previously evaluated designs [Daulton et al., 2021]—similar to the treatment of EI in the single objective setting discussed earlier—and thereby mitigates failures modes of EHVI and qEHVI due to observation noise. EHVI and the noisy and batch variants are quite popular due to their empirical performance [Daulton et al., 2021], and they are one-step Bayes-optimal in their respective settings (e.g. noisy, parallel) when the Pareto set is restricted to be a sub set of the in-sample designs. However, many other HVI-based BO algorithms exist (e.g., Bradford et al. [2018], Konakovic Lukovic et al. [2020]).

3

Robust Multi-Objective Bayesian Optimization Under Input Noise

Contents

3.1 Abstract

Bayesian optimization (BO) is a sample-efficient approach for tuning design parameters to optimize expensive-to-evaluate, black-box performance metrics. In many manufacturing processes, the design parameters are subject to random input noise, resulting in a product that is often less performant than expected. Although BO methods have been proposed for optimizing a single objective under input noise, no existing method addresses the practical scenario where there are multiple objectives that are sensitive to input perturbations. In this work, we propose the first multi-objective BO method that is robust to input noise. We formalize our goal as optimizing the multivariate value-at-risk (MVaR), a risk measure of the uncertain

objectives. Since directly optimizing MVaR is computationally infeasible in many settings, we propose a scalable, theoretically-grounded approach for optimizing MVaR using random scalarizations. Empirically, we find that our approach significantly outperforms alternative methods and efficiently identifies optimal robust designs that will satisfy specifications across multiple metrics with high probability.

3.2 Introduction

Scientists and engineers frequently face optimization problems where the goal is to tune a set of parameters to optimize multiple competing black-box objective functions. Typically, no single design is best with respect to every objective. Hence, the goal in multi-objective (MO) optimization is to identify the Pareto frontier (PF) of optimal trade-offs between the objectives and the corresponding optimal designs. These problems are ubiquitous in a variety of domains including manufacturing [Liao et al., 2008], materials design [Ashby, 2000], robotics [Calandra and Peters, 2014], and machine learning [Izquierdo et al., 2021, Eriksson et al., 2021]. Obtaining measurements of the objectives often requires resource-intensive simulation or experimentation. Therefore, any practical routine for optimizing such functions must be highly sample-efficient; that is, the method must identify optimal design parameters while only querying the objective functions at a small number of designs. Bayesian optimization (BO) [Shahriari et al., 2016] is a popular technique for addressing this class of problems.

In many real-world applications, the final implementation of the selected design parameters are subject to input noise, resulting from uncontrollable perturbations of the input parameters [Beyer and Sendhoff, 2007]. For example, many vaccine production processes involve freeze-drying procedures to stabilize active pharmaceutical ingredients to increase storage lifetime [Mortier et al., 2016, Xie and Schenkendorf, 2019]. Tuning the operating parameters, such as shelf temperature and chamber pressure, can significantly improve the efficiency of the drying step with little reduction in product quality. However, shelf temperature and chamber

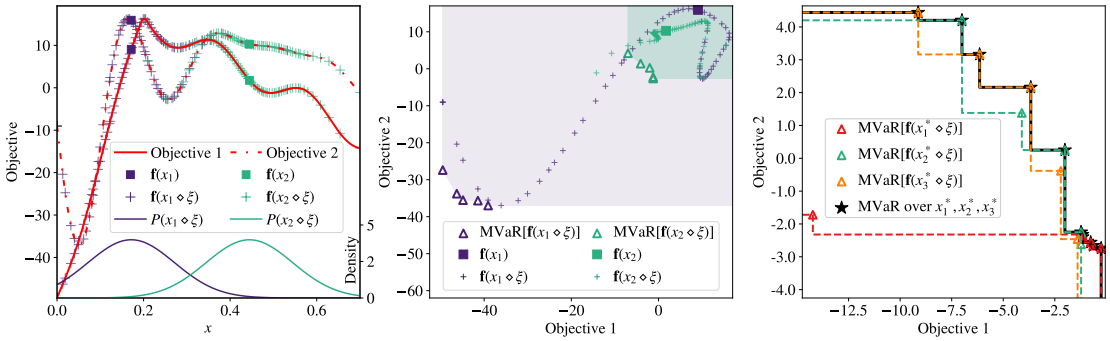

Figure 3.1: A toy problem where the goal is to tune a single parameter x to maximize two objectives. Left: The nominal values for a non-robust (purple, x_1) and a robust design design (green, x_2) are indicated using squares. The plus markers illustrate objective values of each design under zero-mean Gaussian input noise with a standard deviation of 0.1. The non-robust design can lead to low objective values under input perturbations. Center: An illustration of the MVaR sets of the non-robust and robust designs. The triangles represent a discrete approximation of the MVaR set of each design under the input noise distribution. For each point $\boldsymbol{z}$ in the MVaR set of a given design, the objective values for that design subject to input perturbations are $\geq \boldsymbol{z}$ with probability $\alpha \geq 0.9$. In other words, the objectives under input noise for each design will fall in the respective shaded region with probability $\geq \alpha$. Under input noise, the non-robust design x_1 may result in poor objective values despite yielding better values (relative to the robust design x_2) without perturbations. After accounting for input noise, x_2 is a more robust solution than x_1. Right: The MVaR set (black stars) across three optimal designs x_1^*, x_2^*, x_3^* is the set of optimal points across the union of the MVaR sets (colored triangles) of each design.

pressure are subject to uncontrollable random input noise around the *nominal*[1] input parameters. Robustness with respect to this input noise is critical. Higher temperatures lead to greater efficiency, but also make the process more sensitive to perturbations in the temperature because if temperature exceeds the critical collapse threshold, there is irreversible product damage. A more conservative temperature that is robust to input noise may be a better choice than a higher temperature that is worse with respect to the *nominal objectives*. In such high-stakes and high-throughput production pipelines, decision-makers seek to identify robust design parameters that ensure the manufactured products will have high objective quality with high probability.

Optimization without consideration of input noise can lead to solutions that are catastrophic when subjected to input noise at the implementation stage [Doltsinis

[1]We call the design as specified by the decision maker the *nominal design*, and the corresponding value without any perturbations as the *nominal values* for the *nominal objectives*.

and Kang, 2004]. The toy problem that is illustrated in left plot of Figure 3.1 demonstrates a scenario where a non-robust design (depicted as a purple square) is better than the robust design (marked by a green square) with respect to the nominal values. However, the performance metrics for the non-robust design can be significantly worse when the design is subject to input noise. In contrast, the objectives are far less sensitive to input perturbations around the robust design.

To identify designs that are robust to a noisy performance metric due to input noise or observation noise, the value-at-risk (VaR) is often used as the robust objective because it provides high-probability performance guarantees [Basel Committee on Banking Supervision, 2012]. The α VaR is the $(1-\alpha)$-quantile (where $0 \leq \alpha \leq 1$) of the cumulative distribution function (CDF) of the performance metric. Under input noise, variation in the objective is induced via the uncertain inputs. Intuitively, the VaR is the largest value such that the objective value of a given design subject to input perturbations will be greater than that value with probability at least α. Thus, in the context of manufacturing, a design with α VaR exceeding the target specification produces a yield of at least α.

Many recent works have considered BO methods that are robust to input noise in the single objective setting, e.g., by optimizing VaR [Nguyen et al., 2021b] or other risk measures [Fröhlich et al., 2020, Cakmak et al., 2020, Nguyen et al., 2021a]. However, no previous work has considered sample-efficient, generally-applicable multi-objective Bayesian optimization (MOBO) methods that are robust to input perturbations. To our knowledge, the only existing MO methods that are robust to input noise are evolutionary algorithms (EA) that often require tens of thousands of evaluations (e.g., Deb and Gupta [2005]).

In the MO setting, high-probability performance guarantees of a single design can be assessed using the multivariate value-at-risk (MVaR) [Prèkopa, 2012]. MVaR maps a probability value α to a set of vectors where each element provides a lower bound on the the objectives' possible values under input noise with probability α. As illustrated in the center plot of Figure 3.1, the MVaR set of a robust

design (x_2) often provides significantly better probabilistic lower bounds than the MVaR set of a non-robust design (x_1).

Similar to the PF in the standard MO setting—where the PF is the set of optimal trade-offs between objectives with no input noise—the *global* MVaR set is the set of optimal trade-offs that can be achieved with probability $\geq \alpha$ under input noise across all possible designs. The MVaR across a set of 3 designs is illustrated in the right plot of Figure 3.1.

While MVaR provides a natural measure of robust performance guarantees, it is relatively expensive to compute, making it computationally challenging to directly optimize MVaR in the context of MOBO (see Appendix 3.D.1 for a discussion). In this work, we propose a family of novel, theoretically-grounded methods for optimizing MVaR via random scalarizations that mitigates many challenges associated with optimizing MVaR directly.

Contributions

1. We introduce robust MO optimization under input noise, formalize the problem in terms of optimizing global MVaR—a novel, probabilistic form of a robust PF—and discuss computational challenges unique to this setting.
2. We derive a novel theoretical connection between the MVaR and the VaR of a particular scalarization of the objectives, which motivates a family of computationally efficient BO methods for identifying MVaR-optimal designs using an MVaR Approximation based on Random Scalarizations (MARS).
3. We demonstrate that MARS vastly outperforms non-robust alternatives on a variety of synthetic and real-world robust MO optimization problems, including a pharmaceutical manufacturing application.
4. We derive and evaluate extensions of our methods to handle expensive-to-evaluate black-box constraints and parallel candidate generation.

3.3 Background

Multi-Objective Optimization In MO optimization, the goal is to, without loss of generality, maximize a vector-valued black-box function $\max_{\boldsymbol{x}\in\mathcal{X}} \boldsymbol{f}(\boldsymbol{x})$ where $\boldsymbol{f}(\boldsymbol{x}) := [f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})], M \geq 2$, and $\mathcal{X} \subset \mathbb{R}^d$ is a compact search space. Often, there may be additional black-box constraints $\boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}$, where $\boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^V,\ V > 0$, that must be satisfied. We consider the setting where $\boldsymbol{f}$ and $\boldsymbol{c}$ have no known analytic expressions and no known or observed gradients. The goal is to identify the Pareto frontier (PF) of optimal trade-offs and corresponding Pareto set of optimal designs $\mathcal{X}^*$.

Notation For vectors $\boldsymbol{y}, \boldsymbol{y}' \in R^M$. Let $\geq$ and $>$ denote the component-wise extensions of the order notations $\geq$ and $>$, i.e., $\boldsymbol{y} \geq \boldsymbol{y}' \iff y_i \geq y_i'\ \forall\ i$ and $\boldsymbol{y} > \boldsymbol{y}' \iff y_i > y_i'\ \forall\ i$, where $\cdot_i$ denotes the i^{th} element.

Definition 3.3.1. A vector $\boldsymbol{f}(\boldsymbol{x})$ *Pareto dominates* $\boldsymbol{f}(\boldsymbol{x}')$, denoted by $\boldsymbol{f}(\boldsymbol{x}) \succ \boldsymbol{f}(\boldsymbol{x}')$, if $\boldsymbol{f}(\boldsymbol{x}) \geq \boldsymbol{f}(\boldsymbol{x}')$ and $\exists\ j \in \{1, \ldots, M\}$ such that $f^{(j)}(\boldsymbol{x}) > f^{(j)}(\boldsymbol{x}')$.

Definition 3.3.2. The *Pareto frontier* over a set of objective vectors $\mathcal{F} = \{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in X \subseteq \mathcal{X}\}$ is $\textsc{Pareto}(\mathcal{F}) = \{\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{F} \ :\ \nexists\ \boldsymbol{x}' \in X\ s.t.\ \boldsymbol{f}(\boldsymbol{x}') \succ \boldsymbol{f}(\boldsymbol{x})\}$. If there are constraints $\boldsymbol{c}(\boldsymbol{x})$, elements of Pareto are subject to the additional membership assumption that $\boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}$. We call the corresponding set of optimal designs the *Pareto set*.

Although the true PF $\mathcal{P}^* = \textsc{Pareto}(\{\boldsymbol{f}(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{X}})$ is typically unknown, an MO optimization algorithm can be employed to identify a finite approximation. Hypervolume is a commonly used metric to measure the quality of a PF.

Definition 3.3.3. The *hypervolume* (HV) *indicator* of a set of points $\mathcal{Y} \subset \mathbb{R}^M$ is the M-dimensional Lebesgue measure λ_M of the region dominated by $\mathcal{P} := \textsc{Pareto}(\mathcal{Y})$ and bounded from below by a reference point $\boldsymbol{r} \in \mathbb{R}^M$, which we write as $\text{HV}(\mathcal{Y}, \boldsymbol{r})$.

Definition 3.3.4. The *hypervolume improvement* (HVI) of a set of points $\mathcal{Y}'$ with respect to an existing Pareto frontier $\mathcal{P}$ and reference point $\boldsymbol{r}$ is defined as $\text{HVI}(\mathcal{Y}'|\mathcal{P}, \boldsymbol{r}) = \text{HV}(\mathcal{P} \cup \mathcal{Y}'|\boldsymbol{r}) - \text{HV}(\mathcal{P}|\boldsymbol{r})$.[2]

Bayesian Optimization (BO) is a sample-efficient technique for optimizing black-box functions [Frazier, 2018]. BO relies on a probabilistic surrogate model—typically a Gaussian process (GP), which provides well-calibrated uncertainty estimates—and an acquisition function that leverages the surrogate model to quantify the value of evaluating the objective functions for a design $\boldsymbol{x}$. The acquisition function balances exploring areas with high uncertainty and exploiting regions believed to be optimal. BO selects the next point $\boldsymbol{x}$ to evaluate by maximizing the acquisition function (which is cheap to evaluate relative to the objective functions), observes a (potentially noisy) measurement of the metrics $\boldsymbol{y}$, adds the new observation to the dataset $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{y})\} \cup \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, updates the surrogate model to incorporate the new observation, and repeats this process for a predetermined budget of evaluations. In the MO setting, a common approach is to optimize a random scalarization of the objectives using a single objective acquisition function, such as expected improvement [Knowles, 2006] or Thompson sampling [Paria et al., 2020]. Alternatively, a multi-objective acquisition function can be used to directly optimize the PF. For example, expected hypervolume improvement (EHVI) [Emmerich et al., 2006] aims to maximize the HV of the PF under the surrogate model's posterior distribution.

3.4 Related Work

Many recent works on BO have considered settings where at the time of implementation either (i) the parameters are subject to noise [Bogunovic et al., 2018, Oliveira et al., 2019, Fröhlich et al., 2020]—as we consider in this work—or (ii) the system performance depends on auxiliary unknown environmental variables [Kirschner et al., 2020, Iwazaki et al., 2021]. While previous work has focused on

[2]Henceforth, we omit $\boldsymbol{r}$ for brevity. As in previous work, we assume that the reference point $\boldsymbol{r}$ is known and specified by the decision maker [Yang et al., 2019].

optimizing the expected [Toscano-Palmerin and Frazier, 2022] or the worst-case performance [ur Rehman et al., 2014, Sessa et al., 2020], a recent body of work has focused on optimizing more sophisticated risk measures [Picheny et al., 2022, Cakmak et al., 2020, Nguyen et al., 2021a,b]. However, despite recent significant interest in non-robust MOBO [Lukovic et al., 2020, Suzuki et al., 2020], to our knowledge, no prior work has studied robust MOBO.

Motivated by practical limitations due to manufacturing tolerances, Malkomes et al. [2021] proposed constraint active search (CAS), which aims to identify diverse solutions in the region of the search space that exceeds a minimum threshold on the objectives. However, CAS does not model or account for input noise, and CAS alone cannot produce any guarantees on robustness to input noise. Methods such as CAS would require a post-hoc analysis using the data collected during optimization to analyze the sensitivity of the solutions to input noise [Calandra and Peters, 2014].

Approaching robust design by decoupling data collection and sensitivity analysis is central to the Taguchi method [Taguchi, 1989]. Data acquisition often revolves around finding designs that balance the mean and variance of the sensitive objective under input noise [Beyer and Sendhoff, 2007]. Do et al. [2021] propose an approach in this vein for the two-objective setting where only one objective is subject to input noise. However, the algorithm does not seek to identify trade-offs with high probability robustness guarantees, and the method does not handle multiple sensitive objectives. In contrast with the Taguchi method, we aim to unify data collection and sensitivity analysis by selecting designs that are believed to yield high-probability performance guarantees.

Outside the BO literature, robust MO optimization has been studied using either EAs [Gupta and Deb, 2005, He et al., 2019] or assuming access to the explicit mathematical programming formulation of the problem [Majewski et al., 2017, Roberts et al., 2018]. Those works have focused on finding the Pareto frontier of the expectation or the worst-case objectives or on finding the Pareto frontier of the nominal objectives with additional constraints on the deviation from the nominal values [Deb and Gupta, 2005, Avigad and Branke, 2008]. Some works

have considered conceptual properties of different scalarization methods [Ide and Köbis, 2014], but not in relation to MVaR. EAs that are robust to input noise are not applicable to the small evaluation budget regime that we consider because they typically require a large number of function evaluations [Deb and Gupta, 2005]. Even methods that combine EAs with GPs require thousands of evaluations per design [Zhou et al., 2018].

As a final differentiator from prior work, we consider the practical setting where there are additional black-box constraints that are sensitive to input noise [Marzat et al., 2013, Li and Li, 2015], which is a subject addressed by only a few BO methods [Beland and Nair, 2017] even in the single objective case.

3.5 Multi-Objective Optimization with Noisy Inputs

In many practical scenarios, the nominal performance of a design can be evaluated by means of a simulation (e.g., by simulating the pharmaceutical process under nominal operating conditions). We consider the setting where we can simulate $\boldsymbol{f}(\boldsymbol{x})$ for any given design $\boldsymbol{x} \in \mathcal{X}$, but that the design is subject to noise $\boldsymbol{\xi}(\boldsymbol{x})$ from a known noise process $\boldsymbol{\xi}(\boldsymbol{x}) \sim P(\boldsymbol{\xi}; \boldsymbol{x})$ at implementation time.[3] The realized system performance is given by the random variable $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$, where $\boldsymbol{x} \diamond \boldsymbol{\xi}$ denotes any known function $g(\boldsymbol{x}, \boldsymbol{\xi})$ (e.g. for additive noise $\diamond$ is simply $+$). For an extended problem formulation including black-box constraints, see Appendix 3.A.4.

In robust optimization, the goal is often to optimize a *risk measure* $\rho[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})]$ that maps a random variable to a statistic of its distribution. A common risk measure is the expectation over the input noise distribution [Deb and Gupta, 2005, Toscano-Palmerin and Frazier, 2022, Fröhlich et al., 2020], $\mathbb{E}_{\boldsymbol{\xi} \sim P(\boldsymbol{\xi})}[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})]$, which can be used instead of the random variable $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ and optimized via standard multi-objective optimization methods. We propose the first MOBO methods for optimizing expectation objectives in Appendix 3.F. Despite its widespread use, the expectation risk measure may not always align with the practitioner's true

[3] For brevity, we omit the dependency on $\boldsymbol{x}$ in our notation and write $\boldsymbol{\xi}$ and $P(\boldsymbol{\xi})$ going forward.

robustness goals. Often, one would prefer solutions with objectives that are better than some performance specification $\boldsymbol{z} \in \mathbb{R}^M$ with high probability (e.g. to maximize production yield) [Sarykalin et al., 2008]. Hence, in the single-objective setting, probabilistic risk measures such as value-at-risk (VaR) are frequently used.

Definition 3.5.1. Given input noise $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$ where $\boldsymbol{\xi} \in \mathbb{R}^d$ and a confidence level $\alpha \in [0, 1]$, the value-at-risk for a given point $\boldsymbol{x}$ is:

$$\text{VaR}_\alpha\Big[f(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] = \sup\{z \in \mathbb{R} : P\Big[f(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq z\Big] \geq \alpha\}.$$

Although several BO methods exist for optimizing VaR [Cakmak et al., 2020, Nguyen et al., 2021b], they cannot directly be used in the MO setting because VaR is not defined for multivariate random variables. A näive way to extend VaR to the MO setting would be to consider the VaR of each objective independently. However, this ignores the fact that all M objectives are evaluated at the same realization of $\boldsymbol{x} \diamond \boldsymbol{\xi}$. Considering the VaR of each objective independently typically leads to overly optimistic risk estimates because objectives under input noise are not typically *simultaneously* greater than or equal to their respective independent VaRs (i.e. the $(1 - \alpha)$ - quantiles) with probability $\geq \alpha$. Thus it is important to use risk measures such as multivariate value-at-risk (MVaR) that account for the joint distribution of the objectives [Prèkopa, 2012].

This is illustrated in the center plot in Figure 3.1. Under input noise, the objective values are correlated, which underscores the importance of accounting for the joint distribution of the objectives in measures of robustness. In addition, the center plot in Figure 3.1 shows that $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ has an asymmetric distribution, even for this very simple and well-behaved toy problem, which highlights how applying VaR to each objective independently or using the expectation risk measure may conceal underlying variation and risk. Indeed, the results in Appendix 3.I show that optimizing an expectation risk measure on this problem results in poor performance.

In contrast with VaR and the expectation risk measure, which map a random variable to *single* scalar or vector, MVaR maps a random variable to a non-dominated *set* of vectors in the outcome space that are dominated by α-fraction of

all possible realizations, where $\alpha \in [0, 1]$ is a hyperparameter set by the practitioner. That is, each vector in the MVaR set corresponds to an objective specification that a design will meet with probability $\geq \alpha$. Therefore, α is an interpretable risk level that can be valuable in manufacturing applications where one wishes to find the PF of all objective specifications with a guaranteed yield fraction (α).

Definition 3.5.2. The MVaR of $\boldsymbol{f}$ for a given point $\boldsymbol{x}$ and confidence level $\alpha \in [0, 1]$ is:

$$\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] = \text{Pareto}\Big(\Big\{\boldsymbol{z} \in \mathbb{R}^M : P\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}\Big] \geq \alpha\Big\}\Big).$$

The MVaR set over X specifies objective vectors $\boldsymbol{z}$ such that there exists a known design $\boldsymbol{x} \in X$ with corresponding random objectives $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ under $P(\boldsymbol{\xi})$ that dominate $\boldsymbol{z}$ with probability $\geq \alpha$.

Definition 3.5.3. The MVaR for a set of points X is:

$$\text{MVaR}_\alpha\Big[\{\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\}_{\boldsymbol{x} \in X}\Big] = \text{Pareto}\Big(\bigcup_{\boldsymbol{x} \in X} \text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]\Big).$$

The global MVaR across the design space, $\text{MVaR}_\alpha\Big[\{\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\}_{\boldsymbol{x} \in \mathcal{X}}\Big]$, is a robust analogue of the PF in the standard MO setting. The concept of the MVaR of a set of design points X is a novel contribution of this work.

Optimization Goal In this work, our goal is to identify the MVaR set across the design space: $\text{MVaR}_\alpha\Big[\{\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\}_{\boldsymbol{x} \in \mathcal{X}}\Big]$. Given an approximate MVaR set across the design space, a decision-maker can pick a design according to their preferences. Similar to the standard MO setting, the HV of the MVaR set across the design space can be used to evaluate optimization performance.

3.6 Optimizing MVaR

A natural approach for optimizing MVaR is to directly maximize the HV dominated by the MVaR set. Although MVaR of a given point typically cannot be

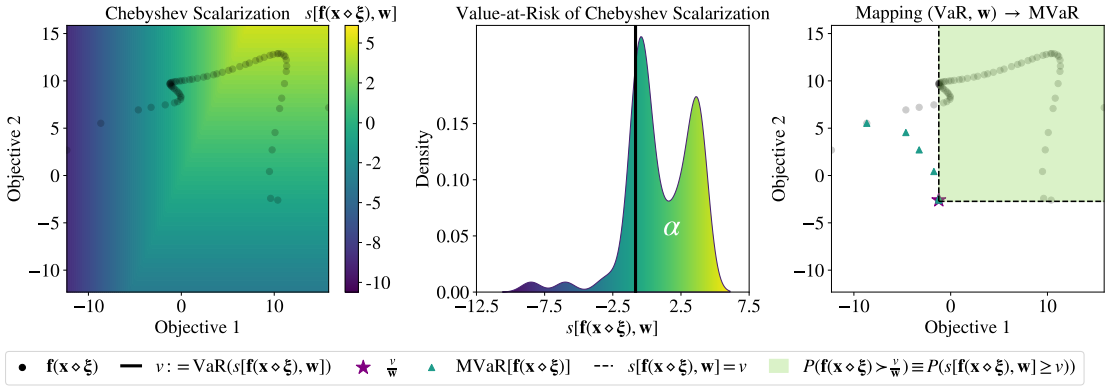

Figure 3.2: Construction of MVaR sets via random scalarizations for the 1-d example in Figure 3.1. Left: The function values for a single design with zero-mean Gaussian input perturbations with a standard deviation of 0.1 are marked by black points. The background is a contour showing the value of a Chebyshev scalarization across the objective space. Center: The probability density of a Chebyshev scalarization over the function values under input noise and the value-at-risk v of a Chebyshev scalarization for $\alpha = 0.9$. The probability mass to the right of the black line is equal to α. Right: Leveraging Theorem 3.6.1, $\mathrm{VaR}_{\boldsymbol{w}}$ can be mapped to point in the MVaR set that is dominated by the objectives under input perturbations 90% of the time. The green triangles represent a discrete approximation of the MVaR set with 64 samples from the input noise distribution. The green area indicates the region that dominates the identified MVaR point, or, equivalently, the area for which the Chebyshev scalarization defined by $\boldsymbol{w}$ is greater than v.

evaluated directly, it can be approximated using $n_{\boldsymbol{\xi}}$ MC samples of $\boldsymbol{\xi}$, provided that independent samples can be drawn from the noise process. Thus, evaluating the MVaR set across the previously evaluated designs using the surrogate requires sampling from the posterior of $P(\boldsymbol{f}|\mathcal{D})$ evaluated jointly at $\boldsymbol{x}_1 \diamond \boldsymbol{\xi}_i, \ldots, \boldsymbol{x}_n \diamond \boldsymbol{\xi}_i$ for $i = 1, ..., n_{\boldsymbol{\xi}}$, where $X_{1:n} := \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ are the previously evaluated designs. Since $\{\boldsymbol{f}(\boldsymbol{x}' \diamond \boldsymbol{\xi})\}_{\boldsymbol{x}' \in X_{1:n}}$ is typically not observed, the corresponding posterior predictions may have large uncertainties. In order to get a reliable estimate of MVaR, we would need to integrate over the posterior distribution of $\{\boldsymbol{f}(\boldsymbol{x}' \diamond \boldsymbol{\xi})\}_{\boldsymbol{x}' \in X_{1:n}}$. qNEHVI [Daulton et al., 2021] is a variant of EHVI that integrates over the uncertainty in function values at previously evaluated designs. This makes qNEHVI suitable for optimizing MVaR.

However, several computational issues—including time complexity that is exponential in the number of objectives and exponential in the size of $\mathrm{MVaR}_\alpha\left[\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi})\right]$—make it infeasible to directly optimize MVaR with qNEHVI in many settings. We

defer a detailed discussion to Appendix 3.D and present an empirical evaluation in Appendix 3.I.

3.6.1 Relationship between MVaR and Scalarizations

An alternative to direct optimization of the MVaR set is to apply a scalarization to the objectives and use a standard risk measure on the scalarized objective. Unlike the use of independent risk measures on each objective, this approach accounts for the correlation between outcomes induced by the input perturbation. In this section, we present our main theoretical result: under limited assumptions, there exists a bijection, based on VaR, that maps a particular family of scalarizations—Chebyshev scalarizations [Kaisa, 1999]—to points in the MVaR set. In other words, each point in the MVaR set corresponds to a particular set of scalarization weights. This means that we can recover the entire MVaR set using these scalarizations, without any loss. Proofs and additional theoretical results including extensions to the constrained setting are provided in Appendix 3.A.

Definition 3.6.1. Let $\boldsymbol{w} \in \Delta_+^{M-1}$, where Δ_+^{M-1} denotes the positive $(M-1)$-simplex, and let $\boldsymbol{r} \in \mathbb{R}^M$. The Chebyshev scalarization $s[\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{r}]$ is given by $s[\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{r}] = \min_i w_i(y_i - r_i)$, where $\cdot_i$ denotes the i^{th} dimension.[4]

The contour in the left plot in Figure 3.2 shows the Chebyshev scalarization for a fixed $\boldsymbol{w}$ for the two-objective toy problem from Figure 3.1 and illustrates a connection between Pareto dominance and the Chebyshev scalarization, which we formalize below. The black points are function values under sampled perturbations for a single design $\boldsymbol{x}$. The center plot in Figure 3.2 shows the distribution of Chebyshev scalarization values for a given $\boldsymbol{w}$ and the black line indicates the α-level VaR. The right plot in Figure 3.2 illustrates that using the VaR of a Chebyshev scalarization, we can deduce a point $\boldsymbol{z}$ such that the function values under the input perturbations will dominate $\boldsymbol{z}$ with probability $\geq \alpha$.

[4]Typically, $\boldsymbol{f}$ is scaled to the unit cube using the $\boldsymbol{r}$ as the lower bound before applying the scalarization. Since the scaled reference point is $\mathbf{0}$, hence forth, we omit $\boldsymbol{r}$ for brevity. See Appendix 3.G.1 for details.

Lemma 3.6.1 (VaR of Chebyshev scalarization ⇒ Pareto Dominance)**.** *Let* $v = \mathit{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$*. Then,* $P\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \frac{v}{\boldsymbol{w}}\Big] \geq \alpha$*, where* $\frac{v}{\boldsymbol{w}}$ *denotes element-wise division.*

The condition in Lemma 3.6.1 is one criterion for membership in the MVaR set (the other being Pareto efficiency). The shaded region in right plot in Figure 3.2 illustrates the region that dominates $\boldsymbol{z}$. With probability $\geq \alpha$, $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ will fall within the shaded region. *This result enables translating the VaR of a Chebyshev scalarization into an interpretable, high-probability guarantee on robust performance in terms of Pareto dominance.*

Assumption 3.6.1. $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ has a continuous, strictly increasing CDF F. I.e., if $\boldsymbol{f}(\boldsymbol{x}) \succ \boldsymbol{f}(\boldsymbol{x}')$, then $F\Big[\boldsymbol{f}(\boldsymbol{x})\Big] > F\Big[\boldsymbol{f}(\boldsymbol{x}')\Big]$.[5]

If Assumption 3.6.1 is met, then for any $\boldsymbol{w}$, $\frac{v}{\boldsymbol{w}}$ is not dominated by any other point in the MVaR set, and hence, $\frac{v}{\boldsymbol{w}}$ is an element of the MVaR set. Furthermore, we have the following:

Theorem 3.6.1 (MVaR $\iff$ VaR of Chebyshev scalarization)**.** *Given* $\boldsymbol{x}, \boldsymbol{f}$*, and* $P(\boldsymbol{\xi})$*, let* $h : \mathit{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] \to \Delta_+^{M-1}$ *be the function* $h(\boldsymbol{z}) = \boldsymbol{w} = \frac{1}{\boldsymbol{z}||\frac{1}{\boldsymbol{z}}||}$*. Under Assumption 3.6.1,* $h(\cdot)$ *is bijective and* $h^{-1}(\boldsymbol{w}) = \boldsymbol{z} = \frac{1}{\boldsymbol{w}}\mathit{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$*.*

Theorem 3.6.1 *provides a technique for generating points in the MVaR set using the VaR of Chebyshev scalarizations with different weights.* Importantly, any design that is globally optimal with respect to MVaR is a maximizer of the VaR of a Chebyshev scalarization. This naturally motivates a methodology for identifying the global MVaR set by optimizing the VaR of random Chebyshev scalarizations.

Corollary 3.6.1 (Consistent Optimizers)**.** *Suppose* $\boldsymbol{z}$ *is a point in the global MVaR set, i.e.,* $\boldsymbol{z} \in \mathit{MVaR}_\alpha\Big[\{\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\}_{\boldsymbol{x} \in \mathcal{X}}\Big]$*. Let* $\mathcal{X}_{\boldsymbol{z}}^*$ *be the set of designs such that for all* $\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{z}}^*$*,* $\boldsymbol{z} \in \mathit{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$*. Then every* $\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{z}}^*$ *is a maximizer of* $\mathit{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$ *for* $\boldsymbol{w} = \Big(\boldsymbol{z}||\frac{1}{\boldsymbol{z}}||\Big)^{-1}$*.*

[5] Assumption 3.6.1 holds when $\boldsymbol{f}$ is a function sampled from a GP prior with many commonly used covariance functions. See Lemma 3.A.5 for a formal statement and Appendix 3.A.3 for proof.

3.6.2 MARS: MVaR Approximation via Random Scalarizations

The connection between the VaR of a Chebyshev scalarization and MVaR can be exploited to optimize the global MVaR by randomly sampling a Chebyshev scalarization at each BO iteration[6] and optimizing the VaR of the Chebyshev scalarization using a single objective BO algorithm, such as Noisy Expected Improvement (NEI) [Letham et al., 2019]—which is required (rather than expected improvement) since we must integrate over the unknown best unknown incumbent value—or Thompson sampling (TS) [Thompson, 1933, Paria et al., 2020]. We refer to this technique as MVaR Approximation via Random Scalarizations (MARS). MARS is a simple, theoretically-grounded technique, and we find, in Section 3.7, that MARS performs well empirically. In the main text, we focus on MARS with NEI (denoted as MARS-NEI) , but we derive and empirically evaluate upper confidence bound (UCB) [Srinivas et al., 2010] and TS variants in Appendices 3.B and 3.G.

For MARS-NEI, we use the MC formulation of NEI [Balandat et al., 2020] so that the Chebyshev scalarizations can be computed in the NEI as composite objectives [Astudillo and Frazier, 2019]. We optimize the the acquisition function using sample-path gradients that leverage well-studied gradient estimators of VaR (see Appendix 3.C.1 for details). See Appendix 3.G.1 for details on optimization.

3.7 Experiments

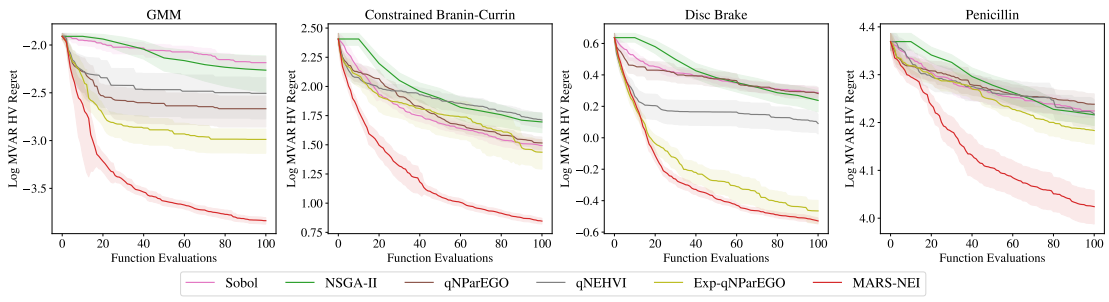

Figure 3.3: The log MVaR HV regret after the initial space-filling design. For each method, we plot the mean and 2 standard errors of the mean over 20 trials.

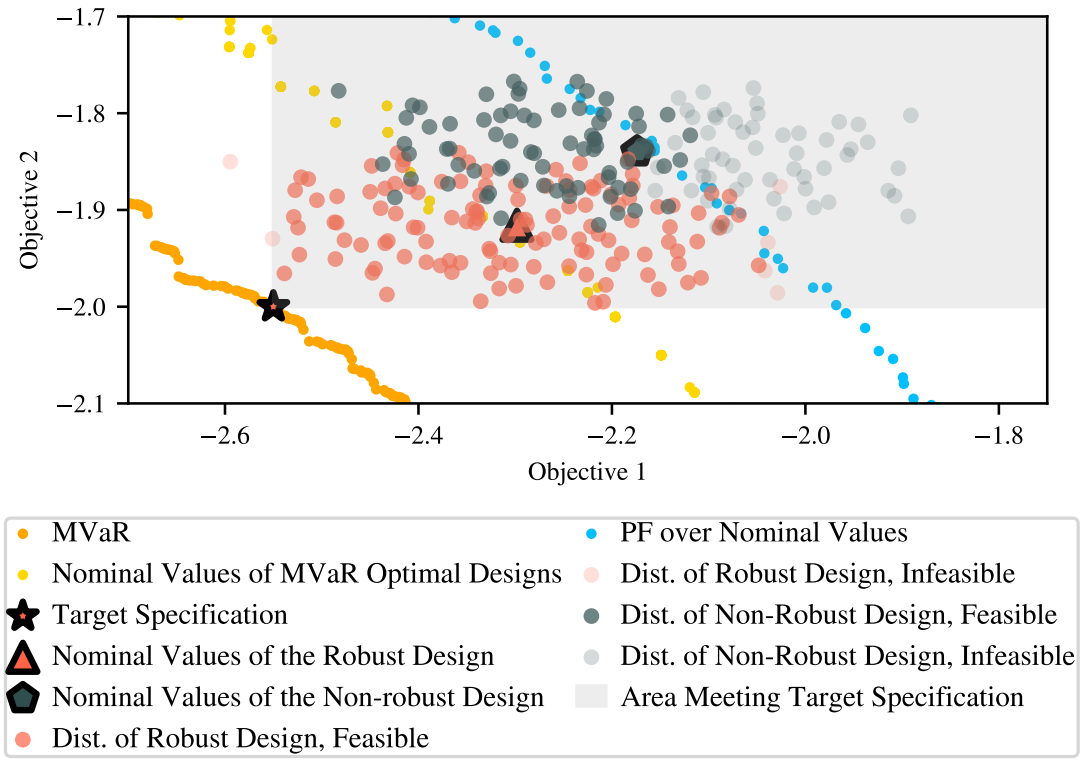

Figure 3.4: The yield from selecting a robust versus a non-robust design on the Disc Brake problem. Although the non-robust design is feasible under the nominal objectives, it is located near the boundary of the feasible region in design space and violates some of the black-box constraints (not shown) under a large fraction of input perturbations.

In this section, we provide an empirical demonstration of robust MOBO on synthetic and real-world problems. We compare three broad classes of methods: (i) Non-robust methods including NSGA-II [Deb et al., 2002], qNParEGO [Daulton et al., 2020], qNEHVI [Daulton et al., 2021], (ii) methods for optimizing expectation risk measures, e.g. using qNParEGO (denoted as Exp-qNParEGO), and (iii) methods for optimizing MVaR via MARS. For readability, we only include one expectation and one MVaR optimization method in the main text, both based on NEI. In Appendix 3.G, we evaluate additional methods including MARS with TS and UCB and methods for direct MVaR optimization based on NEHVI. We consider the expectation risk measure because it is simple and performs well in many scenarios. All robust (expectation and MVaR) methods are our novel contributions.

[6]We sample weights uniformly from Δ_+^{M-1}, which we find works well empirically.

Additionally, we compare against a quasi-random policy, which selects the designs to evaluate according to a scrambled Sobol sequence [Owen, 2003].

For all BO-methods, we begin by evaluating $2(d+1)$ design points from a scrambled Sobol sequence. We use $n_{\boldsymbol{\xi}} = 32$ samples because we find that setting $n_{\boldsymbol{\xi}} > 32$ yields little-to-no improvement in optimization performance (see Figure 3.I.7). See Appendix 3.G for details on all methods. Our code is open-sourced at `github.com/facebookresearch/robust_mobo`.

3.7.1 Synthetic Problems

Gaussian Mixture Model (GMM) ($d = 2,\ M = 2,\ \alpha = 0.9$): This is a variant of the GMM problem from Fröhlich et al. [2020] where each objective is an independent GMM. We use a multiplicative noise model, i.e., $\boldsymbol{x} \diamond \boldsymbol{\xi} := \boldsymbol{x}\boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(\mu = \mathbf{1}, \Sigma = 0.07 I_2)$ with I_n denoting the n-dimensional identity matrix. In Appendix 3.I.5, we present multiple variations of this problem to demonstrate the consistency of our methods under different noise models.

Constrained Branin Currin ($d = 2,\ M = 2,\ V = 1,\ \alpha = 0.7$): We subject this problem, which originates from Daulton et al. [2020], to a heteroskedastic input noise process given by $P(\boldsymbol{\xi}; \boldsymbol{x}) = \mathcal{N}\left(\mu = \mathbf{0}, \Sigma = 0.05\left(1 + \sigma\left(1 - 2x_0\right)\right) I_2\right)$, where $\sigma(x) = \frac{1}{1+e^{-x}}$. The optimal designs with respect to the nominal objectives are on the boundary of the feasible region with respect to the outcome constraint. Hence, the optimal designs with respect to the nominal metrics often violate the constraint under input perturbations.

3.7.2 Real-World Problems

Disc Brake ($d = 4,\ M = 2,\ V = 4,\ \alpha = 0.95$): In this disc brake manufacturing problem, the goal is to minimize the brake's mass and the stopping time of a vehicle by tuning the inner and outer radii of the disc, the engaging force, and the number of friction surfaces [Ray and Liew, 2002]. Following Emch and Parkinson [1994], we use zero-mean uniform input noise with a maximum absolute perturbation

value of 5% of the range of each parameter (except for the number of friction surfaces, which is noise-free).

Penicillin Production ($d = 7,\ M = 3,\ \alpha = 0.8$): This problem considers optimizing the manufacturing process of penicillin [Liang and Lai, 2021]. The goal is to maximize the yield while minimizing time-to-ferment and CO_2 output by tuning 7 initial conditions of the chemical reaction. Each parameter is subject to independent zero-mean Gaussian noise, where the standard deviation ranges from 0.5% to 3% of the parameter's domain (see Appendix 3.G for details).

3.7.3 Results

In Figure 3.I.4, we evaluate all methods in terms of log HV regret, which is the difference in HV between the true global MVaR set and the MVaR of the set of designs evaluated by each method (see Appendix 3.G.3 for details on estimating true global MVaR). Figure 3.3 shows that MARS is consistently the best performing method. The non-robust methods consistently perform poorly on all problems. On the GMM problem, Exp-qNParEGO performs worse than MARS-NEI because the optimal designs under expectation risk measure are in a disjoint part of the search space from the MVaR-optimal designs. On the Constrained Branin Currin problem, the nominal methods perform no better than quasi-random Sobol search. On the penicillin problem, MARS vastly outperforms all other methods.

Although the log HV regret highlights the performance of MARS, it does not fully capture the *necessity* of using a robust method in practice. In Figure 3.4, we analyze the yield (i.e. the probability of the objectives exceeding a performance specification under the input noise distribution) of a robust and a non-robust design on the Disc Brake problem. Using a target performance specification chosen from the MVaR set, we see that if a decision-maker were to select a non-robust design (green pentagon) that is optimal with respect to the nominal objectives and nominal values that meet the target specification, the yield would only be 58.2%. In contrast, an MVaR-optimal solution can be chosen such that it meets target specification with high probability. For example, the robust design marked by the

orange triangle enjoys a yield of 95.3%. As shown in Figure 3.4, this is because the non-robust design often does not satisfy all of the black-box constraints under input perturbations. In this problem, the objectives are relatively robust to noise (much more so than the toy example in Figure 3.1), but the feasibility of a design (and therefore the yield) is highly sensitive to input noise when the design is near the boundary of the feasible region in design space.

Table 3.H.1 reports the wall times for running a single iteration of BO with each algorithm. Not only is MARS-NEI computationally tractable on all problems, but it achieves wall times that are competitive to alternative algorithms.

Evaluation in additional problem settings and comparisons against additional methods in Appendix 3.I further validate that MARS-NEI is consistently a top performer and yields competitive wall times. We find that MARS-TS performs slightly worse, on average, in terms of log HV regret than MARS-NEI, but that it has shorter wall times. Methods for direct MVaR optimization with qNEHVI perform comparably to MARS-NEI, but the direct MVaR optimization is prohibitively expensive in terms of wall time and memory requirements and was infeasible to run on many problems (including nearly all problems with > 2 objectives). In contrast, Figure 3.I.2 shows that MARS-NEI can scale to problems with > 2 objectives and consistently performs best in those settings. Additionally, in Appendix 3.G, we show that MARS-NEI works well under a wide variety of input noise processes, scales well with increasing batch sizes (in the parallel evaluation setting), and is not sensitive to the number of MC samples $n_{\boldsymbol{\xi}}$ used to estimate $P(\boldsymbol{\xi})$, for $n_{\boldsymbol{\xi}} \geq 32$.

3.8 Discussion

In this work, we formulate the goal of MO robust optimization under input noise as optimizing the *global* MVaR set—a novel concept that is a robust analogue of the Pareto frontier in the standard MO setting. We derive a correspondence between MVaR and Chebyshev scalarizations based on VaR. This theoretical result naturally motivates a computationally efficient approach (MARS) for using BO to optimize

the global MVaR set with high sample-efficiency. Empirically, we find that MARS consistently outperforms alternative approaches and achieves competitive wall times.

Although our focus has been on the small evaluation budget regime and BO, our theoretical results are far more general. The connection between MVaR and Chebyshev scalarizations could be leveraged by gradient-based and evolutionary methods to scale global MVaR optimization to settings with large evaluation budgets. We hope that our contributions serve as a foundation for future advances in methods for robust MO optimization under input noise.

Appendix

AppendixAppendices

3.A Theory and Proofs

3.A.1 Preliminaries

Definition 3.A.1. Let $\mathcal{F} = \{\boldsymbol{f}(\boldsymbol{x}) : \boldsymbol{x} \in X \subseteq \mathcal{X}\}$. The weakly efficient Pareto frontier is $\textsc{WeakPareto}(\mathcal{F}) = \{\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{F} \ : \ \nexists \ \boldsymbol{x}' \in X \ s.t. \ \boldsymbol{f}(\boldsymbol{x}') > \boldsymbol{f}(\boldsymbol{x})\}$, and $\textsc{Pareto}(\mathcal{F}) \subseteq \textsc{WeakPareto}(\mathcal{F})$. If there are constraints $\boldsymbol{c}(\boldsymbol{x})$, elements of WeakPareto are subject to the additional membership Assumption that $\boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}$. We call the corresponding set of optimal designs the *weak Pareto set*.

Definition 3.A.2. Weak-MVaR is defined in the same way as MVaR, but only requires that its elements are weakly Pareto optimal.

Definition 3.A.3 (Prèkopa [2012])**.** If Assumption 3.6.1 holds, then we can express MVaR with an equality with respect to α:

$$\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] = \Big\{\boldsymbol{z} \in \mathbb{R}^M : P\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}\Big] = \alpha\Big\}. \tag{3.1}$$

3.A.2 Proofs

Lemma 3.A.1. *Let $\boldsymbol{y} \in \mathbb{R}^M$ and $v \in \mathbb{R}$. Then $s[\boldsymbol{y}, \boldsymbol{w}] \geq v \iff \boldsymbol{y} \geq \frac{v}{\boldsymbol{w}}$.*

Proof. This follows directly from Definition 3.6.1.

$$\begin{aligned} s[\boldsymbol{y}, \boldsymbol{w}] \geq v &\iff \min_i w_i y_i \geq v \\ &\iff w_i y_i \geq v \ \forall i \\ &\iff y_i \geq \frac{v}{w_i} \ \forall i \\ &\iff \boldsymbol{y} \geq \frac{v}{\boldsymbol{w}}. \end{aligned}$$

□

Lemma 3.A.1 states that a lower bound v on the value of a Chebyshev scalarization of an objective vector $\boldsymbol{y}$ can be used to define a point, $\frac{v}{\boldsymbol{w}}$, that $\boldsymbol{y}$ is greater than or equal to. We can extend this to make a similar statement about the VaR of a Chebyshev scalarization.

Lemma 3.6.1 (VaR of Chebyshev scalarization ⇒ Pareto Dominance)**.** *Let* $v = \mathit{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$*. Then,* $P\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \frac{v}{\boldsymbol{w}}\Big] \geq \alpha$*, where* $\frac{v}{\boldsymbol{w}}$ *denotes element-wise division.*

Proof. From Definition 3.5.1, we have that

$$\mathrm{VaR}_\alpha\Big[f(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] = \sup\{z \in \mathbb{R} : P\Big[f(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq z\Big] \geq \alpha\}.$$

Hence, $P(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}] \geq v) \geq \alpha$. From Definition 3.6.1, we have $P(\min_i[w_i f_i(\boldsymbol{x} \diamond \boldsymbol{\xi})] \geq v) \geq \alpha$. By Lemma 3.A.1, the statement $\min_i[w_i f_i(\boldsymbol{x} \diamond \boldsymbol{\xi})] \geq v$ is equivalent to $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \frac{v}{\boldsymbol{w}}$. Hence, $P\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \frac{v}{\boldsymbol{w}}\Big) \geq \alpha$. □

Lemma 3.A.2 (VaR of Chebyshev scalarization ⇒ Weak-MVaR)**.** *Let* $v = \mathit{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$*. Then,* $\frac{v}{\boldsymbol{w}} \in \mathit{Weak\text{-}MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$*.*

Proof. Let $\boldsymbol{z} = \frac{v}{\boldsymbol{w}}$. Suppose there exists $\boldsymbol{z}' \in \text{Weak-MVaR}_\alpha(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}))$ such that $\boldsymbol{z}' > \boldsymbol{z}$. Since $\boldsymbol{z}' \in \text{Weak-MVaR}_\alpha(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}))$, we have that $P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}') \geq \alpha$. Note that $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}'$ implies that $\min_i w_i f_i(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \min_i w_i z'_i$. Hence, $P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}') \geq \alpha$ implies that $P(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}] \geq s[\boldsymbol{z}', \boldsymbol{w}]) \geq \alpha$. Since $\boldsymbol{z}' > \boldsymbol{z}$ and $\boldsymbol{w} \in \Delta_+^{M-1}$, we have that $s[\boldsymbol{z}', \boldsymbol{w}] > s[\boldsymbol{z}, \boldsymbol{w}] = v$. But this contradicts Definition 3.5.1. Since there does not exist $\boldsymbol{z}' \in \text{Weak-MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$ such that $\boldsymbol{z}' > \boldsymbol{z}$, we have that $\boldsymbol{z} \in \text{Weak-MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$. □

If Assumption 3.6.1 holds, the inequalities with respect to α in Lemma 3.6.1 become equalities, and we show that $\frac{v}{\boldsymbol{w}}$ is strictly Pareto optimal.

Lemma 3.A.3 (VaR of Chebyshev scalarization ⇒ MVaR)**.** *Let* $v = \mathit{VaR}_\alpha\Big[s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big]$*. If Assumption 3.6.1 holds, then* $\frac{v}{\boldsymbol{w}} \in \mathit{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$*.*

Proof. Let $\boldsymbol{z} := \frac{v}{\boldsymbol{w}}$. Suppose there exists $\boldsymbol{z}' \in \text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$ such that $\boldsymbol{z}' \succ \boldsymbol{z}$. Because $F(\cdot)$ is a strictly increasing CDF, $P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}') = F(\boldsymbol{z}') > F(\boldsymbol{z}) = P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z})$. From Lemma 3.6.1, we have that $P\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}\Big) \geq \alpha$. Because $\boldsymbol{z}' \in \text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$ from Equation (3.1), we have that $P\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}'\Big) = \alpha$. Hence, $P\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}\Big) \geq P\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}'\Big)$. This is a contradiction. □

Lemma 3.A.3 provides a technique for generating points in the MVaR set using the VaR of Chebyshev scalarizations with different $\boldsymbol{w}$.

Now, consider the reverse mapping.

Lemma 3.A.4 (MVaR ⇒ VaR of Chebyshev scalarization)**.** *Suppose that* $\boldsymbol{z} \in MVaR_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$. *Then* $\boldsymbol{z} = \frac{v}{\boldsymbol{w}}$ *for* $\boldsymbol{w} := \frac{1}{\boldsymbol{z}}||\frac{1}{\boldsymbol{z}}||_1^{-1} \in \Delta_+^M$ *and* $v = VaR_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$.

Proof. Without loss of generality, assume that $\boldsymbol{z} > \boldsymbol{0}$.[7] Let $v = ||\frac{1}{\boldsymbol{z}}||^{-1} \in \mathbb{R}_+$. Then, $\boldsymbol{z} = \frac{v}{\boldsymbol{w}}$. Hence, all we need to show is that v equals $\text{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$. Let us define $v' := \text{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$. By definition,

$$v' = \sup\{v'' \in \mathbb{R} : P\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}] \geq v''\Big] \geq \alpha\}.$$

Since $\boldsymbol{z} \in \text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$, $P\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \boldsymbol{z}\Big] \geq \alpha$. Hence, $P\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \frac{v}{\boldsymbol{w}}\Big] \geq \alpha$. Using Lemma 3.A.1, we have that $P\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}] \geq v\Big) \geq \alpha$. Since v' is the supremum, we have that $v \leq v'$. Suppose now that $v < v'$. Note that $\frac{v}{\boldsymbol{w}} = \boldsymbol{z} \in \text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$. From Lemma 3.A.2, $\frac{v'}{\boldsymbol{w}} \in \text{Weak-MVaR}_\alpha\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big)$. Since $v < v'$, $\boldsymbol{z} = \frac{v}{\boldsymbol{w}} \prec \frac{v'}{\boldsymbol{w}}$. Hence, $\boldsymbol{z}$ cannot be in $\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$ because it is dominated by $\frac{v'}{\boldsymbol{w}}$ and $\frac{v'}{\boldsymbol{w}} \in \text{Weak-MVaR}_\alpha\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big)$. This is a contradiction. Hence $v \geq v'$ and therefore it follows that $v = v' = \text{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$. □

Let $h : \text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] \to \Delta_+^{M-1}$ be given by $h(\boldsymbol{z}) = \frac{1}{\boldsymbol{z}||\frac{1}{\boldsymbol{z}}||}$. Being the element-wise application of a scalar injective mapping ($z \mapsto 1/z$), it is clear that h is injective. However, $h(\cdot)$ is not necessarily bijective, since without Assumption 3.6.1 there may be weight vectors $\boldsymbol{w}$ such that $\nexists\ \boldsymbol{z} \in \text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$ s.t . $h(\boldsymbol{z}) = \boldsymbol{w}$.

[7] This can easily be guaranteed by shifting the objectives to be strictly positive. If the objectives are not unbounded from below, the reference point, which is commonly supplied by the decision-maker in multi-objective optimization effectively provides a lower bound on the objectives.

Theorem 3.6.1 (MVaR $\iff$ VaR of Chebyshev scalarization)**.** *Given* $\boldsymbol{x}, \boldsymbol{f}$, *and* $P(\boldsymbol{\xi})$, *let* $h : MVaR_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] \to \Delta_+^{M-1}$ *be the function* $h(\boldsymbol{z}) = \boldsymbol{w} = \frac{1}{\boldsymbol{z}||\frac{1}{\boldsymbol{z}}||}$. *Under Assumption 3.6.1,* $h(\cdot)$ *is bijective and* $h^{-1}(\boldsymbol{w}) = \boldsymbol{z} = \frac{1}{\boldsymbol{w}} VaR_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big)$.

Proof. This follows directly from Lemma 3.A.3 and Lemma 3.A.4. □

Corollary 3.A.1 (MVaR via Scalarization)**.** *Weak-MVaR enjoys the following scalarization representation:* $\text{Weak-}MVaR_\alpha\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big) = \{\frac{1}{\boldsymbol{w}} VaR_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big) \ : \ \boldsymbol{w} \sim \Delta_+^{M-1}\}$. *If Assumption 3.6.1 holds, then* $MVaR_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] = \{\frac{1}{\boldsymbol{w}} VaR_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big) \ : \ \boldsymbol{w} \sim \Delta_+^{M-1}\}$.

Although the MVaR representation in Corollary 3.A.1 depends on Assumption 3.6.1, *Lemma 3.6.1 recovers all weakly Pareto optimal points even if this assumption is not met* because MVaR $\subseteq$ Weak-MVaR. Hence, with or without Assumption 3.6.1, Theorem 3.A.1 can be used to approximate the MVaR set.

Result 3.A.1 (MVaR Approximation)**.** MVaR can be approximated with a finite set of weight vectors: $\widehat{\text{MVaR}}\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big) = \{\frac{1}{\boldsymbol{w}_i} \text{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}_i]\Big)\}_{i=1}^{N_{\text{MVaR}}}$.

3.A.3 Discussion of the Assumption of Continuous, Strictly-Increasing CDFs with Gaussian Processes

In Bayesian Optimization with Gaussian Process surrogates, it is assumed that the objective function $\boldsymbol{f}$ is sample path from a Gaussian process prior. In this setting, the random variable $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ for a deterministic sample path $\boldsymbol{f}$ (where the stochasticity comes solely from $\boldsymbol{\xi}$) has a continuous, strictly increasing CDF $F(\cdot)$ for many commonly used covariance functions. Hence, the bijective relationship in Theorem 3.6.1 holds given a suitable choice of covariance function.[8]

Lemma 3.A.5. *If* $\boldsymbol{f}$ *is a sample path from a multi-output Gaussian Process prior where the covariance function and the covariance function of the derivative process are strictly positive definite and with sample paths that are differentiable,*[9] $\mathcal{X}$ *is a*

[8] Using a discrete MC approximation of the input noise distribution means that the CDF is not strictly increasing.

[9] Sample paths of many commonly used covariance functions are differentiable [Paciorek, 2003].

compact set, and $P(\boldsymbol{x} \diamond \boldsymbol{\xi})$ is a continuous distribution with strictly positive support, then Assumption 3.6.1 holds.

Proof. Consider the case of a scalar function f. By Lemma 1 of Cakmak et al. [2020], the density of $f(\boldsymbol{x} \diamond \boldsymbol{\xi})$ is strictly positive. Hence, any interval with positive Lebesgue measure has non-zero density. So the cumulative density function of $f(\boldsymbol{x} \diamond \boldsymbol{\xi})$ is strictly increasing. Consider the case of a multi-output sample path $\boldsymbol{f}$. Suppose that the joint CDF is not strictly increasing. Then there exist $\boldsymbol{y}, \boldsymbol{y}'$ such that $\boldsymbol{y} \geq \boldsymbol{y}'$ and there exists at least one $i \in \{1, ..., M\}$ *s.t.* $y_i > y'_i$ and $F(\boldsymbol{y}) \leq F(\boldsymbol{y}')$. Since $\boldsymbol{y} \geq \boldsymbol{y}'$ and F is a CDF, $F(\boldsymbol{y}) \geq F(\boldsymbol{y}')$. Hence, $F(\boldsymbol{y}) = F(\boldsymbol{y}')$. So, we have $P(f_1(\boldsymbol{x} \diamond \boldsymbol{\xi}) \leq y_1, ..., f_M(\boldsymbol{x} \diamond \boldsymbol{\xi}) \leq y_M) = P(f_1(\boldsymbol{x} \diamond \boldsymbol{\xi}) \leq y'_1, ..., f_M(\boldsymbol{x} \diamond \boldsymbol{\xi}) \leq y'_M)$. Suppose $y'_i = y_i$ for all $i \neq j$. Then, $P(f_1(\boldsymbol{x} \diamond \boldsymbol{\xi}) \leq y_1, y'_j < f_j(\boldsymbol{x} \diamond \boldsymbol{\xi}) \leq y_j, ..., f_M(\boldsymbol{x} \diamond \boldsymbol{\xi}) \leq y_M) = 0$. But the hyperrectangle bounded by $[-\infty, ..., y'_j, ..., -\infty]$ and $\boldsymbol{y}$ has positive Lebesgue measure. Since the pdf of each of $f_1, ..., f_M$ is strictly positive, the cumulative density over the hyperrectangle is greater than zero, which is a contradiction. The argument is easily extended to the case when there exists $1 \geq k \geq M$ indices $i_1, ..., i_k$ such that $y'_{i_k} < y_{i_k}$. □

3.A.4 Extension to Black-Box Constraints Under Input Noise

In this section, we consider the setting where in addition to the objective function $\boldsymbol{f}$ there is a vector-valued black-box function $\boldsymbol{c}(\boldsymbol{x}) \in \mathbb{R}^V$ specifying the outcome constraint $\boldsymbol{c}(\boldsymbol{x}) > \boldsymbol{0}$ that is also subject to input noise $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$. To handle black-box constraints under input noise, we weight the objectives $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ by a feasibility indicator $\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}]$ that is 1 if all constraints are satisfied and 0 otherwise. We define VaR and MVaR for feasibility-weighted objectives and extend the theoretical results from Section 3.6.1.

The proofs for the results in the constrained setting follow the proofs in Appendix 3.A.2 with slight modifications. 1) Assumption 3.6.1 does not hold for feasibility-weighted objectives. Hence, the implication is that a random sampled scalarization is only guaranteed to correspond to a point in the Weak-MVaR set,

but importantly any point in the MVaR set does correspond to some scalarization and therefore can be recovered. 2) The proofs handle the special case where some of the constraints are not satisfied and the feasibility-weighted objectives are zero.

Definition 3.A.4. The value-at-risk of the feasibility-weighted objective for a given point $\boldsymbol{x}$ is:

$$\text{VaR}_\alpha\Big(f(\boldsymbol{x}\diamond\boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}\diamond\boldsymbol{\xi})>\mathbf{0}]\Big)=\sup\Big\{z\in\mathbb{R}:P\Big(f(\boldsymbol{x}\diamond\boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}\diamond\boldsymbol{\xi})>\mathbf{0}]\geq z\Big)\geq\alpha\Big\}.$$

Definition 3.A.5. The MVaR of the feasibility-weighted objectives $\boldsymbol{f}$ for a given point $\boldsymbol{x}$ is:

$$\begin{aligned}\text{MVaR}_\alpha\Big(\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi}),\boldsymbol{c}(\boldsymbol{x}\diamond\boldsymbol{\xi})\Big)=\Big\{&\\ &\boldsymbol{z}\in\mathbb{R}^M\ s.t.\\ &\quad P\Big(\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}\diamond\boldsymbol{\xi})>\mathbf{0}]\geq\boldsymbol{z}\Big)\geq\alpha,\\ &\quad \nexists\ \boldsymbol{z}'\in\mathbb{R}^M,\boldsymbol{z}'\succ\boldsymbol{z},P\Big(\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}\succ\mathbf{0}]\geq\boldsymbol{z}'\Big)\geq\alpha\\ \Big\}.&\end{aligned}$$

Let $\mathcal{M}^{\boldsymbol{c}}_{\boldsymbol{\xi},X}:=\bigcup_{\boldsymbol{x}\in X}\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi}),\boldsymbol{c}(\boldsymbol{x}\diamond\boldsymbol{\xi})\Big]$. The *global* MVaR of the feasibility weighted objectives for a set of points X is defined as

$$\text{MVaR}_{P(\boldsymbol{\xi}),\alpha}\Big(\{\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi}),\boldsymbol{c}(\boldsymbol{x}\diamond\boldsymbol{\xi})\}_{\boldsymbol{x}\in X}\Big)=\Big\{\boldsymbol{z}\in\mathcal{M}^{\boldsymbol{c}}_{\boldsymbol{\xi},X}:\nexists\ \boldsymbol{z}'\in\mathcal{M}^{\boldsymbol{c}}_{\boldsymbol{\xi},X}\ s.t.\ \boldsymbol{z}'\succ\boldsymbol{z}\Big\}.$$

Weak-MVaR of the feasibility-weighted objectives is defined in the same way, but only requires that its elements are weakly Pareto optimal

Lemma 3.A.6. *Given a weight vector $\boldsymbol{w}\in\Delta_+^{M-1}$, $\boldsymbol{y}\in\mathbb{R}^M$, $\boldsymbol{y}_c\in\mathbb{R}^{M'}$ and $v\in\mathbb{R}$,*

$$s[\boldsymbol{y},\boldsymbol{w}]\mathbb{1}[\boldsymbol{y}_c\succ\mathbf{0}]\geq v\iff\boldsymbol{y}\mathbb{1}[\boldsymbol{y}_c\succ\mathbf{0}]\geq\frac{v}{\boldsymbol{w}}.$$

Proof. This follows directly from Definition 3.6.1.

$$\begin{aligned}s[\boldsymbol{y},\boldsymbol{w}]\mathbb{1}[\boldsymbol{y}_c\succ\mathbf{0}]\geq v&\iff\mathbb{1}[\boldsymbol{y}_c\succ\mathbf{0}]\min_i w_iy_i\geq v\\&\iff w_iy_i\mathbb{1}[\boldsymbol{y}_c\succ\mathbf{0}]\geq v\ \forall i\\&\iff y_i\mathbb{1}[\boldsymbol{y}_c\succ\mathbf{0}]\geq\frac{v}{w_i}\ \forall i\\&\iff\boldsymbol{y}\mathbb{1}[\boldsymbol{y}_c\succ\mathbf{0}]\geq\frac{v}{\boldsymbol{w}}.\end{aligned}$$

□

Theorem 3.A.1 (VaR of Feasibility-Weighted Chebyshev scalarization ⇒ Pareto Dominance)**.** *Given a weight vector* $\boldsymbol{w} \in \Delta_+^{M-1}$*, let* $v = \mathit{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}]\Big)$*. Then,*

$$P\bigg(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] \geq \frac{v}{\boldsymbol{w}}\bigg) \geq \alpha.$$

Proof. From Definition 3.A.4, we have that

$$\mathrm{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}]\Big) = \sup\Big\{z \in \mathbb{R} : s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] \geq z\Big) \geq \alpha\Big\}.$$

Hence,

$$P(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] \geq v) \geq \alpha.$$

From Definition 3.6.1, we have

$$P(\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] \min_i w_i f^{(i)}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq v) \geq \alpha.$$

By Lemma 3.A.6, the following expressions are equivalent

$$\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] \min_i w_i f^{(i)}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq v \iff \boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] \geq \frac{v}{\boldsymbol{w}}.$$

Hence, $P\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] \geq \frac{v}{\boldsymbol{w}}\Big) \geq \alpha$. □

Lemma 3.A.7. *Let* $\boldsymbol{z} \in \mathit{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$*. If* $\boldsymbol{f}(\boldsymbol{x}) > \boldsymbol{0}$*, then* $\boldsymbol{z} \not\succ \boldsymbol{0}$ *if and only if* $\boldsymbol{z} = \boldsymbol{0}$*.*

Proof. The following shows that, since $\boldsymbol{f}(\boldsymbol{x}) > \boldsymbol{0}$, $\boldsymbol{z} \not\succ \boldsymbol{0}$ if and only if $P(\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}) < \alpha$. Thus, $\boldsymbol{z} \not\succ \boldsymbol{0}$ if and only if $\boldsymbol{z} = \boldsymbol{0}$.

Since $\boldsymbol{f}(\boldsymbol{x}) > \boldsymbol{0}$, we have that $z_i \geq 0$ for all $i = 1, ..., M$ and there exists $j \in \{1, ..., M\}$ such that $z_j = 0$. Note that since $\boldsymbol{f}(\boldsymbol{x}) > \boldsymbol{0}$, the i^{th} element $f^{(i)}(\boldsymbol{x})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}) > \boldsymbol{0}] = 0$ if and only if $\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}) > \boldsymbol{0}] = 0$. Hence, either $\boldsymbol{f}(\boldsymbol{x})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}) > \boldsymbol{0}] > \boldsymbol{0}$ or $\boldsymbol{f}(\boldsymbol{x})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}) > \boldsymbol{0}] = \boldsymbol{0}$.

From Definition 3.A.5, we have that $P\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] \geq \boldsymbol{z}\Big) \geq \alpha$. Since $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] > \boldsymbol{0}$ or $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] = \boldsymbol{0}$,

$$P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] > \boldsymbol{0}) + P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \boldsymbol{0}] = \boldsymbol{0}) = 1.$$

Suppose $\boldsymbol{z} \not\succ \mathbf{0}$. Since $\boldsymbol{z}$ is not dominated by any other point in the MVaR set, $P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}] \succ \mathbf{0}) < \alpha$. Hence, $\boldsymbol{z}$ must be $\mathbf{0}$. □

Lemma 3.A.8 (VaR of Feasibility-Weighted Chebyshev scalarization ⇒ Feasibility-Weighted Weak-MVaR)**.** *Let* $v = \mathit{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}], \boldsymbol{w}]\Big)$*. Then,* $\frac{v}{\boldsymbol{w}} \in \mathit{Weak\text{-}MVaR}_\alpha\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big)$.

Proof. Without loss of generality, assume that the objectives $\boldsymbol{f}(\boldsymbol{x}) > \mathbf{0}$.[0] Let $\boldsymbol{z} = \frac{v}{\boldsymbol{w}}$. Suppose there exists $\boldsymbol{z}' \in \text{Weak-MVaR}_\alpha(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}))$ such that $\boldsymbol{z}' > \boldsymbol{z}$. Since $\boldsymbol{z}' \in \text{Weak-MVaR}_\alpha(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}))$,

$$P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}] \geq \boldsymbol{z}') \geq \alpha.$$

Note that $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}] \geq \boldsymbol{z}'$ implies that

$$\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}] \min_i w_i f^{(i)}(\boldsymbol{x} \diamond \boldsymbol{\xi}) \geq \min_i w_i z_i.$$

Hence, $P(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}] \geq \boldsymbol{z}') \geq \alpha$ implies that

$$P(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}] \geq s[\boldsymbol{z}', \boldsymbol{w}]) \geq \alpha.$$

Note that $s[\boldsymbol{z}', \boldsymbol{w}] > s[\boldsymbol{z}, \boldsymbol{w}] = v$. But by the Definition 3.A.4, v is the maximum value such that

$$P(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}] \geq v) \geq \alpha.$$

This is a contradiction. □

Theorem 3.A.2 (Feasibility-Weighted MVaR ⇒ VaR of Feasibility-Weighted Chebyshev scalarization)**.** *For any* $\boldsymbol{z} \in \mathit{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$*, there exists* $\boldsymbol{w} \in \Delta_+^{M-1}$ *such that* $\boldsymbol{z} = \frac{v}{\boldsymbol{w}}$ *where* $v = \mathit{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}]\Big)$.

Proof. Without loss of generality, assume that the objectives $\boldsymbol{f}(\boldsymbol{x}) > \mathbf{0}$.[0]

Case 1: $\boldsymbol{z} \not\succ \mathbf{0}$. From Lemma 3.A.7, we have that $\boldsymbol{z} = \mathbf{0}$. Note that since the MVaR set contains only non-dominated points and $\mathbf{0}$ is a lower bound on $\boldsymbol{f}(\boldsymbol{x})\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}) > \mathbf{0}]$, $\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] = \{\mathbf{0}\}$. Let $v := \text{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}]\Big)$.

Suppose $v = 0$. Then, for any $\boldsymbol{w} \in \Delta_+^{M-1}$, $\frac{v}{\boldsymbol{w}} = \mathbf{0} \in \text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$.

Suppose $v > 0$. Then for any $\boldsymbol{w} \in \Delta_+^{M-1}$, $\frac{v}{\boldsymbol{w}} > \mathbf{0}$. By Lemma 3.A.8, $\frac{v}{\boldsymbol{w}} \in \text{Weak-MVaR}_\alpha\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big)$. But $\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] = \{\mathbf{0}\}$, and any point in the MVaR set is non-dominated. So $\frac{v}{\boldsymbol{w}} \not\succ \mathbf{0}$. This is a contradiction.

Hence, $v = 0$. Therefore, Theorem 3.A.2 holds when $\boldsymbol{z} \not\succ \mathbf{0}$.

Case 2: $\boldsymbol{z} \succ \mathbf{0}$.

Consider the vector $\frac{1}{\boldsymbol{z}}$. If we divide $\frac{1}{\boldsymbol{z}}$ by its L1-norm, we obtain a vector $\boldsymbol{w} := \frac{1}{\boldsymbol{z}}||\frac{1}{\boldsymbol{z}}||_1^{-1} \in \Delta_+^M$, where $||\cdot||_1$ denotes the L1-norm. Let $v = \frac{1}{||\frac{1}{\boldsymbol{z}}||_1} \in \mathbb{R}_+$. Then, $\boldsymbol{z} = \frac{v}{\boldsymbol{w}}$. Hence, all we need to show is that $v = \text{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}]\Big)$. By definition,

$$\text{VaR}_\alpha\Big(s[f(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}]\Big) = \sup\Big\{v'' \in \mathbb{R} : P\Big(s[f(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}] \geq v''\Big) \geq \alpha\Big\}.$$

Let us define $v' := \text{VaR}_\alpha\Big(s[f(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}]\Big)$. Suppose that $v < v'$. By Lemma 3.A.8, $\frac{v'}{\boldsymbol{w}} \in \text{Weak-MVaR}_\alpha\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big)$. Since $v < v'$, $\boldsymbol{z} = \frac{v}{\boldsymbol{w}} \prec \frac{v'}{\boldsymbol{w}}$. But $\boldsymbol{z}$ is in $\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$. So by Definition 3.5.2, $\boldsymbol{z}$ is not dominated by any other vector in $\text{Weak-MVaR}_\alpha\Big(\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big)$. This is a contradiction. Hence, $v = \text{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}]\Big)$. Thus, Theorem 3.A.2 holds when $\boldsymbol{z} \succ \mathbf{0}$. □

Corollary 3.A.2. *There is a injective function* $g : MVaR_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] \to \Delta_+^{M-1}$ *such that* $g(\boldsymbol{z}) = \frac{1}{\boldsymbol{z}||\frac{1}{\boldsymbol{z}}||} = \boldsymbol{w}$ *and* $\boldsymbol{z} = \frac{v}{\boldsymbol{w}}$ *where* $v = VaR_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\Big) = \frac{1}{||\frac{1}{\boldsymbol{z}}||}$.

Corollary 3.A.3. *Suppose* $\boldsymbol{z}$ *is a point in the global MVaR set* $\boldsymbol{z} \in MVaR_\alpha\Big[\{\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\}_{\boldsymbol{x} \in \mathcal{X}}\Big]$. *Let* $\mathcal{X}_{\boldsymbol{z}}^*$ *be the set of designs such that for all* $\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{z}}^*$, $\boldsymbol{z} \in MVaR_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big]$. *Then every* $\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{z}}^*$ *is a maximizer of* $VaR_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}]\Big)$.

When $\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] \neq \{\mathbf{0}\}$, it follows directly from the injective mapping from $\boldsymbol{z}$ to $\boldsymbol{w}$ that $\text{VaR}_\alpha\Big(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]\mathbb{1}[\boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi}) > \mathbf{0}]\Big)$ is the same for all $\boldsymbol{x} \in \mathcal{X}_{\boldsymbol{z}}^*$. When $\text{MVaR}_\alpha\Big[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{c}(\boldsymbol{x} \diamond \boldsymbol{\xi})\Big] = \{\mathbf{0}\}$, then all $\boldsymbol{x}$ are infeasible designs and $\mathcal{X}_{\boldsymbol{z}}^* = \mathcal{X}$.

3.B MARS with Alternative Acquisition Functions

In this section, we discuss using MARS with two alternative acquisition functions: Thompson Sampling (TS) and Upper Confidence Bound (UCB).

3.B.1 MARS with Thompson Sampling

As discussed in Section 3.6, direct MVaR optimization with qNEHVI requires evaluating the joint posterior over $n_{\boldsymbol{\xi}}(n+1)$ designs. The same is true when using MARS-NEI. Although low-rank Cholesky updates can significantly reduce the complexity [Osborne and of Oxford, 2010], further computational improvements can be obtained by using TS with random Fourier features (RFFs) [Rahimi and Recht, 2008]. However, RFFs are approximate GP samples and introduce approximation error (see Appendix 3.D.3 for further discussion).[10] We refer to this method as MARS-TS. MARS-TS naturally supports (i) parallel candidate generation by drawing a new posterior sample and new scalarization weights for each candidate; (ii) constraints, by evaluating the feasibility-weighted objectives under the posterior sample, and (iii) noisy observations.

3.B.2 MARS with Upper Confidence Bound

Another computationally efficient approach is to use Upper Confidence Bound (UCB), which does not require the expensive integration over $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ where $\boldsymbol{x} \in X_{1:n}$. We refer to this method as MARS-UCB. In what follows, we show how to extend the V-UCB algorithm of Nguyen et al. [2021b] to optimize the VaR of Chebyshev scalarizations. The result builds on the following lemma by Chowdhury and Gopalan [2017], which holds under the assumption that the function $f^{(i)}(\cdot)$ belongs to a reproducing kernel Hilbert space (RKHS) $\mathcal{F}_{k_i}(B_i)$, whose RKHS norm is bounded by $\|f^{(i)}\|_{k_i} \leq B_i$, where $\boldsymbol{f} = [f^{(1)}, ..., f^{(M)}]$. We use $\mu_n^{(i)}(x), \Sigma_n^{(i)}(x,x)$ to denote the posterior, conditional on observations up to iteration n, mean and

[10] Alternative approaches for efficient posterior sampling such as decoupled sampling [Wilson et al., 2020] could also be used.

variance of the GP surrogate corresponding to i^{th} objective, and use σ_i^2 to denote the observation noise for the i^{th} objective.

Lemma 3.B.1. *Chowdhury and Gopalan [2017]. For $\delta \in (0,1)$, $\zeta_{n+1}^{(i)} = B_i + \sigma_i^2\sqrt{2(\gamma_n + 1 + \log(1/\delta))}$, the following holds for all $\boldsymbol{x} \in \mathcal{X}$ with probability $\geq 1-\delta$:*

$$l_n^{(i)}(\boldsymbol{x}) \leq f^{(i)}(\boldsymbol{x}) \leq u_n^{(i)}(\boldsymbol{x}), \tag{3.2}$$

where $l_n^{(i)}(\boldsymbol{x}) := \mu_n^{(i)}(\boldsymbol{x}) - \zeta_{n+1}^{(i)}(\Sigma_n^{(i)}(\boldsymbol{x},\boldsymbol{x}))^{1/2}$, $u_n^{(i)}(\boldsymbol{x}) := \mu_n^{(i)}(\boldsymbol{x}) + \zeta_{n+1}^{(i)}(\Sigma_n^{(i)}(\boldsymbol{x},\boldsymbol{x}))^{1/2}$, and γ_n denotes the maximum information gain.

Assuming that each objective is modeled using an independent GP surrogate and considering all objectives jointly, we see that (3.2) holds jointly for all $i = 1,\ldots,m$ with probability at least $(1-\delta') := (1-\delta)^m$. Applying the Chebyshev scalarization,

$$\begin{aligned} w_i l_n^{(i)}(\boldsymbol{x}) \leq w_i f^{(i)}(\boldsymbol{x}) \leq w_i u_n^{(i)}(\boldsymbol{x}), \forall i = 1,\ldots,M \\ \min_i w_i l_n^{(i)}(\boldsymbol{x}) \leq \min_i w_i f^{(i)}(\boldsymbol{x}) \leq \min_i w_i u_n^{(i)}(\boldsymbol{x}) \\ s[\boldsymbol{l}_n(\boldsymbol{x}), \boldsymbol{w}] \leq s[\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}] \leq s[\boldsymbol{u}_n(\boldsymbol{x}), \boldsymbol{w}] \end{aligned}$$

holds with probability $\geq (1-\delta')$ for all $\boldsymbol{x} \in \mathcal{X}$. Following Lemma 2 of Nguyen et al. [2021b], we get that

$$\text{VaR}_\alpha(s[\boldsymbol{l}_n(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]) \leq \text{VaR}_\alpha(s[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}]) \leq \text{VaR}_\alpha(s[\boldsymbol{u}_n(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}])$$

holds with probability at least $(1-\delta')$. Thus, the UCB policy for VaR of Chebyshev scalarization is defined as the policy that samples $\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x}} \text{VaR}_\alpha(s[\boldsymbol{u}_n(\boldsymbol{x} \diamond \boldsymbol{\xi}), \boldsymbol{w}])$.

In practice, computing the $\zeta_{n+1}^{(i)}$ given in Lemma 3.B.1 is impractical, and typically leads to an acquisition function that is overly conservative. Thus, we follow Nguyen et al. [2021b] and use $\zeta_{n+1}^{(i)} = 2\log(n^2\pi^2/0.6)$ in the experiments.

MARS-UCB can be extended to support parallel candidate generation by sampling a new scalarization for each candidate and noisy observations. The UCB policy derived above does not hold for feasibility-weighted objectives because Lemma 3.B.1 requires that $f^{(i)}(\cdot)$ belongs to a RKHS, and this is not the case when $f^{(i)}(\cdot)$ is weighted by a feasibility indicator because it is no longer continuous [de Freitas et al., 2012].

3.C Gradient-based Acquisition Function Optimization

3.C.1 Approximate Gradients of VaR

One of the earliest and simplest-to-use gradient estimators for VaR was presented by Hong [2009]. Under mild regularity assumptions on the distribution of the random variable, they establish the consistency of the VaR gradient estimator, which can be seen as the sample-path gradient of the well-known estimator of VaR. For this discussion, let $g(\cdot)$ be a deterministic function of its argument (e.g., a sample path of the GP), let us fix $\boldsymbol{x}$, and let $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$ be a continuous random variable. Let $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_k$ denote i.i.d. samples from $P(\boldsymbol{\xi})$. Define the following ordering of the samples, where the subscript $(\cdot)$ denote the order statistic:

$$g(\boldsymbol{x} \diamond \boldsymbol{\xi}_{(1)}) \leq g(\boldsymbol{x} \diamond \boldsymbol{\xi}_{(2)}) \leq \ldots \leq g(\boldsymbol{x} \diamond \boldsymbol{\xi}_{(k)}).$$

The VaR at risk level α can be estimated by

$$\text{VaR}_{\boldsymbol{\xi} \sim P(\boldsymbol{\xi})}(g(\boldsymbol{x} \diamond \boldsymbol{\xi})) \approx g(\boldsymbol{x} \diamond \boldsymbol{\xi}_{(\lfloor (1-\alpha)k \rfloor)}),$$

where $\lfloor \cdot \rfloor$ denotes the largest integer less than or equal to $\cdot$. It is well known (cf. Serfling [2008]) that this estimator is consistent as $k \to \infty$. Hong [2009] extend this result to show that the corresponding gradient estimator

$$\nabla_{\boldsymbol{x}} \text{VaR}_{\boldsymbol{\xi} \sim P(\boldsymbol{\xi})}(g(\boldsymbol{x} \diamond \boldsymbol{\xi})) \approx \nabla_{\boldsymbol{x}} g(\boldsymbol{x} \diamond \boldsymbol{\xi}_{(\lfloor (1-\alpha)k \rfloor)})$$

is an asymptotically (as $k \to \infty$) unbiased estimator of the gradient of VaR. The estimator is also consistent as long as $\nabla_{\boldsymbol{x}} g(\boldsymbol{x} \diamond \boldsymbol{\xi}_{(\lfloor (1-\alpha)k \rfloor)})$ is not a function of ξ, otherwise, averaging of multiple sample gradients is required to obtain a consistent estimator of the gradient of VaR.

In addition to the sample-path gradient estimator discussed above, there are other estimators of gradients of VaR that are based on, e.g., the likelihood ratio gradient estimation or on the kernel density estimators. A detailed discussion of these can be found in Hong et al. [2014].

3.C.2 Approximate Gradients of MVaR

Differentiability of MVaR, more precisely the differentiability of the elements of the MVaR set, is a subject that has not been explored in the literature. Since the computation of the MVaR set corresponding to a set of posterior samples is expensive enough to be the bottleneck during acquisition function optimization, it is highly desirable to avoid the finite-difference gradient estimation, which requires multiple evaluations of the objective and is in general less efficient than the sample-path gradients. Instead, it is preferable to establish a direct connection between the MVaR set and the gradients of the samples on which the MVaR is computed. The method we discuss below is inspired by the gradients of VaR, which correspond to the gradients of the sample that is equal to VaR.

The correspondence between $\nabla_{\boldsymbol{x}}\text{VaR}_{\boldsymbol{\xi}\sim P(\boldsymbol{\xi})}(g(\boldsymbol{x}\diamond\boldsymbol{\xi}))$ and $\nabla_{\boldsymbol{x}}g(\boldsymbol{x}\diamond\boldsymbol{\xi}_{(\lfloor(1-\alpha)k\rfloor)})$ follows from the observation that, since $g(\cdot)$ is a continuous function, shifting $\boldsymbol{x}$ by a sufficiently small $\boldsymbol{\epsilon}$ should not change the ordering of $\boldsymbol{\xi}$'s. We should still have $\text{VaR}_{\boldsymbol{\xi}\sim P(\boldsymbol{\xi})}(g(\boldsymbol{x}+\boldsymbol{\epsilon}+\boldsymbol{\xi}))\approx g(\boldsymbol{x}+\boldsymbol{\epsilon}+\boldsymbol{\xi}_{(\lfloor(1-\alpha)k\rfloor)})$ with the same ordering, as long as $g(\boldsymbol{x}\diamond\boldsymbol{\xi}_{(i)})\neq g(\boldsymbol{x}\diamond\boldsymbol{\xi}_{(j)})$ for $i\neq j$. The same idea extends to the MVaR. Using a finite set of samples to approximate the MVaR set, with $\boldsymbol{m}(\boldsymbol{x}\diamond\boldsymbol{\xi})$ denoting an arbitrary element of the MVaR set, we have that $m^{(j)}(\boldsymbol{x}\diamond\boldsymbol{\xi})=f^{(j)}(\boldsymbol{x}\diamond\boldsymbol{\xi}_{i^{(j)}})$ for some $i^{(j)}\in\{1,\ldots,k\}$, where $\boldsymbol{f}=[f^{(1)},...,f^{(M)}]$ and the $i^{(j)}$ is dependent on the outcome j and the particular element of the MVaR set. This can be interpreted as saying that the elements of the MVaR set are constructed by piecing together outcomes from the samples of the random variable.

Similar to what was discussed for VaR, if we perturb $\boldsymbol{x}$ by a small $\boldsymbol{\epsilon}$, under the assumption that $f^{(j)}(\boldsymbol{x}\diamond\boldsymbol{\xi}_i)\neq f^{(j)}(\boldsymbol{x}\diamond\boldsymbol{\xi}_k)$ for $i\neq k$, we should get that $m^{(j)}(\boldsymbol{x}+\boldsymbol{\epsilon}+\boldsymbol{\xi})=f^{(j)}(\boldsymbol{x}+\boldsymbol{\epsilon}+\boldsymbol{\xi}_{i^{(j)}})$ with the same $i^{(j)}$ as before the perturbation. This, in essence, says that we can calculate the gradients of the elements of the MVaR set as $\nabla_{\boldsymbol{x}}m^{(j)}(\boldsymbol{x}\diamond\boldsymbol{\xi})=\nabla_{\boldsymbol{x}}f^{(j)}(\boldsymbol{x}\diamond\boldsymbol{\xi}_{i^{(j)}})$. Putting all outcomes together, we get

$$\nabla_{\boldsymbol{x}}\boldsymbol{m}(\boldsymbol{x}\diamond\boldsymbol{\xi})=[\nabla_{\boldsymbol{x}}f^{(1)}(\boldsymbol{x}\diamond\boldsymbol{\xi}_{i^{(1)}}),\ldots,\nabla_{\boldsymbol{x}}f^{(M)}(\boldsymbol{x}\diamond\boldsymbol{\xi}_{i^{(M)}})].$$

A theoretical consistency analysis of these MVaR gradient estimators is beyond the scope of this paper. However, we observe that they do work well in practice, enabling efficient optimization of MVaR-NEHVI-RFF (see Appendix 3.D&3.I).

3.D Direct MVaR Optimization using qNEHVI

In this section, we discuss direct optimization of MVaR using NEHVI, highlight the computational challenges that come with this approach, and introduce an approximation that mitigates some of these challenges for some problems where the MVaR set for a design is relatively small (e.g. where the number of objectives is small and α is large). However, we find that these approaches are typically infeasible when $M \geq 3$ due to GPU memory limits (see for example Table 3.I.1).

3.D.1 Direct Optimization of MVaR with NEHVI

As described in Section 3.6, the extension of qNEHVI to optimize MVaR is conceptually simple. qNEHVI selects the next point to evaluate by maximizing the expected HVI under the GP posterior. We replace the standard HVI of a new point with respect to the PF with the joint HVI of the MVaR set of a new point with respect to the MVaR set over the previously evaluated designs:

$$\alpha_{\text{MVaR-NEHVI}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{f}\sim P(\boldsymbol{f}|\mathcal{D})}\Big[\text{HVI}(\text{MVaR}_\alpha[\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi})] \mid \text{MVaR}_\alpha[\{\boldsymbol{f}(\boldsymbol{x}'\diamond\boldsymbol{\xi})\}_{\boldsymbol{x}'\in X_{1:n}}]\Big].$$

However, there are several computational issues that make this approach prohibitively expensive, except for when the objective evaluation takes multiple hours or days. In our experiments, we observe long runtimes, even when using very reasonable parameter values of $n_{\boldsymbol{\xi}} = 32$ and $\alpha = 0.9$. We discuss the factors contributing to this in the following subsection. Note that all the issues discussed are compounded by the fact that even when using a gradient-based approach to optimize the acquisition function, the acquisition function needs to be evaluated many times.

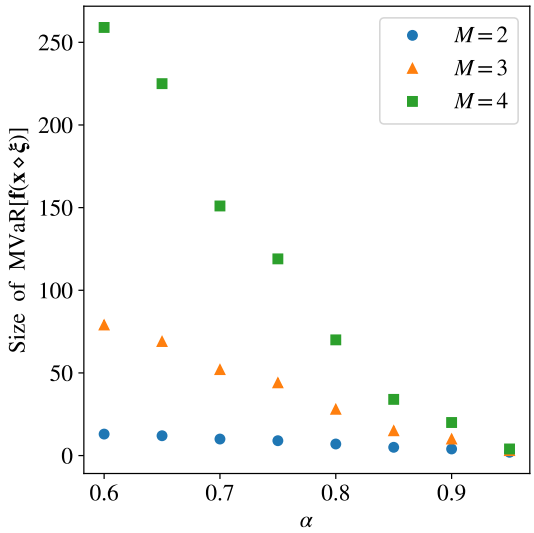

Figure 3.D.1: The maximum size of $\text{MVaR}_\alpha[\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})]$ across the design space $\boldsymbol{x} \in \mathcal{X}$, which is an estimator of the maximum MVaR set size that will encountered during numerical optimization, for different α with $n_{\boldsymbol{\xi}} = 32$ on the GMM problem. The size of the MVaR set significantly increases as α decreases and as M increases.

3.D.2 Complexity and Challenges

There are three primary computational bottlenecks, corresponding to three stages of computing $\alpha_{\text{MVaR-NEHVI}}(\boldsymbol{x})$. We discuss each stage and their complexity below.

1. **Posterior Sampling**: Computing $\alpha_{\text{MVaR-NEHVI}}(\boldsymbol{x})$ requires drawing joint posterior samples at the baseline points (points that are already evaluated) and the current candidate(s) $\boldsymbol{x}$ under all $n_{\boldsymbol{\xi}}$ perturbations. For all methods using GPs, posterior sampling at n points and $n_{\boldsymbol{\xi}}$ perturbations scales as $O(n_{\boldsymbol{\xi}}^3 n^3 M)$. Hence, using GPs is only feasible for modest $n_{\boldsymbol{\xi}}$.

2. **Computing MVaR**: The next stage is computing the MVaR corresponding to each $\boldsymbol{x}$ from posterior samples of $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$. Computing MVaR involves computing the distribution function of $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$, [11] which has a time and space complexity of $\mathcal{O}((1-\alpha)^M n_{\boldsymbol{\xi}}^{M+1} M)$. The MVaR has to be computed for each $\boldsymbol{x}$ and each posterior sample, further inflating the computational effort required.

[11] The distribution of $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ is unknown and the GP posterior over $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ is generally intractable. Hence, the distribution of $\boldsymbol{x} \diamond \boldsymbol{\xi}$ is often approximated with a finite set of MC samples [Cakmak et al., 2020] because the GP posterior can be evaluated analytically over a finite set of points.

The size of the resulting MVaR set is $\mathcal{O}((1-\alpha)^M n_{\boldsymbol{\xi}}^M M)$. Figure 3.D.1 empirically demonstrates that the size of the MVaR set significantly increases as α decreases, particularly for larger M.

3. **Computing joint hypervolume improvement**: Given the samples of MVaR corresponding to the baseline points and the candidate(s), the final step of MVaR-NEHVI is to compute the joint HVI of the MVaR set of the candidates over the global MVaR set corresponding to the baseline points. To our knowledge, the only existing differentiable approach for joint HVI computation relies on the inclusion-exclusion principle (IEP, Daulton et al. [2020]). The time and space complexity of computing the joint HVI of a set of q' points using the IEP is exponential with respect to q'. To compute the HVI of the MVaR for a set of q candidates, we must compute the joint HVI of a set of size $q' = q|\text{MVaR}_\alpha[\{\boldsymbol{f}(\boldsymbol{x}_i + \boldsymbol{\xi})\}_{i=1}^q)]|$. Since the size of the MVaR set of a single candidate scales as $\mathcal{O}((1-\alpha)^M n_{\boldsymbol{\xi}}^M M)$, using IEP quickly becomes infeasible, except for very moderate $M, n_{\boldsymbol{\xi}}$, and α. As shown in Figure 3.D.1, even for $q = 1, n_{\boldsymbol{\xi}} = 32$, and $M = 3$, the size of the MVaR set can be quite large for smaller values of α, which precludes the use of IEP. Although IEP is necessary to make the joint hypervolume improvement computation differentiable, there are non-differentiable approaches (e.g. Lacour et al. [2017]) that could be used instead to compute the joint hypervolume improvement. However, optimizing $\alpha_{\text{MVaR-NEHVI}}$ without gradients would be very slow given that Daulton et al. [2020] showed that simply optimizing analytic EHVI (without MVaR) with CMA-ES [Hansen, 2007] or L-BFGS-B [Byrd et al., 1995] with approximate gradients estimated via finite differences is over an order of magnitude slower than when using exact gradients.

A final challenge in using MVaR-NEHVI is that the calculation of MVaR is not differentiable, and there are no known theoretically-grounded gradient estimators of MVaR. Therefore, we use the heuristic approach described in Appendix 3.C.2 for estimating the gradients of MVaR.

3.D.3 Approximating qNEHVI with RFF Draws

Random Fourier Features (RFF, Rahimi and Recht [2008]) offer an inexpensive and differentiable approximation of GP sample paths. Daulton et al. [2021] propose to combine a single RFF draw with NEHVI to obtain a cheap approximation, qNEHVI-1. They find that qNEHVI-1 is competitive with qNEHVI in small dimensional search spaces, though its performance degrades as the dimensionality increases.

We follow their approach and extend qNEHVI-1 to optimize MVaR, and name this method MVaR-NEHVI-RFF. Using RFFs significantly reduces the computational cost of the evaluating the acquisition function because it avoids computationally expensive exact posterior sampling. In addition, since we use a single RFF draw, the MVaR set and its HVI only have to be computed once per acquisition function evaluation rather than for each posterior sample. In the end, for small problem instances, MVaR-NEHVI-RFF ends up with a per-iteration runtime measured in seconds, which is an immense reduction from the time it takes for MVaR-NEHVI using an exact GP model.

Note that MVaR-NEHVI-RFF still requires the use of IEP for differentiable HVI computations, making this approach infeasible in many settings as discussed above. In addition, the performance of RFFs based acquisition functions are known to degrade when the underlying function is difficult to model and variance starvation is a known issue [Wang et al., 2018, Wilson et al., 2020, Mutny and Krause, 2018, Calandriello et al., 2019]. Thus, for an acquisition function that works well in all settings, we recommend using MARS.

3.E Pruning for Efficient Joint Posterior Sampling

NEI and qNEHVI both require sampling from the joint posterior over function values at the new design $\boldsymbol{x}$ and previously evaluated designs $X_{1:n}$, which we denote by $P(\{\boldsymbol{f}(\boldsymbol{x})\}_{\boldsymbol{x}\in X_{1:n}\cup\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_q\}}|\mathcal{D})$. To reduce the cost of posterior sampling, we prune

$X_{1:n}$ to only include the subset of points $X_{\text{pruned}} \subseteq X_{1:n}$ that have nonzero probability of being optimal. We estimate the probability of being optimal using MC estimation with N_{prune} samples from the joint posterior. For NEI, optimality is with respect to a scalar objective and often means $|X_{\text{pruned}}| << n$. For qNEHVI, any design that has nonzero probability of being Pareto optimal is retained; typically, this results in a much larger X_{pruned} than we using NEI with a scalar objective. The typically larger size of X_{pruned} under qNEHVI-based methods has a significant effect when using MVaR-qNEHVI, where sampling from the joint posterior scales as $\mathcal{O}(Mnn_{\boldsymbol{\xi}})$ and has a very significant effect on runtime. Pruning strategies that leverage techniques from pre-screening and population selection in EAs may further improve computational efficiency.

3.F Optimization of Multi-Objective Expectation Objectives

As noted in the main text, the optimization of expectation of objectives can be achieved via rather straightforward extensions of the existing multi-objective acquisition functions. Here, we discuss the main idea, and show how to extend qNParEGO and qNEHVI [Daulton et al., 2021].

3.F.1 Optimization Expectation Objectives with qNParEGO

The acquisition function ParEGO is an extension of the well known Expected Improvement acquisition function to the multi-objective setting via augmented Chebyshev scalarizations. qNParEGO is an MC-based variant that uses composite objectives with the NEI acquisition function [Daulton et al., 2021]. Given a weight vector $\boldsymbol{w} \in \Delta_+^{M-1}$, it selects the next point to evaluate as follows:

$$\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}_{\boldsymbol{f} \sim P(\boldsymbol{f}|\mathcal{D})} \left[s_a[\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}] - \max_{\boldsymbol{x}' \in X_{1:n}} s_a[\boldsymbol{f}(\boldsymbol{x}'), \boldsymbol{w}] \right]_+, \tag{3.3}$$

where $X_{1:n}$ denotes the points evaluated so far, and $[\cdot]_+$ denotes $\max(\cdot, 0)$, $s_a[\boldsymbol{y}, \boldsymbol{w}] = \min w_i y_i + \beta \sum_i w_i y_i$, and β is a small positive constant. The expectation in (3.3) is not available in closed form, and is typically replaced by a (Q)MC approximation

obtained by drawing samples of $\{\boldsymbol{f}(\boldsymbol{x}')\}_{\boldsymbol{x}'\in X_{1:n}\cup\{\boldsymbol{x}\}}$ from the joint GP posterior. A batch of q candidates can be selected in a sequential greedy fashion where each point is selected using a different scalarization weight vector and the improvement from the batch of q points replaces the improvement from a single point in (3.3).

To extend qNParEGO to the expectation objectives, we replace each occurrence of $s_a[\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{w}]$ in (3.3) with $s_a\left[\mathbb{E}_{\boldsymbol{\xi}\sim P(\boldsymbol{\xi})}[\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi})], \boldsymbol{w}\right]$. For implementation, we follow the same MC idea, and draw samples from the joint posterior of $\{\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi})\}_{\boldsymbol{x}\in X_{1:n}\cup\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_q\},\boldsymbol{\xi}\in\Xi}$ where Ξ is a set of $n_{\boldsymbol{\xi}}$ input noise samples, and approximate (3.3) using these samples. To improve the computational efficiency, one can also calculate the posterior distribution of $\{\mathbb{E}_{\boldsymbol{\xi}\sim P(\boldsymbol{\xi})}[\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi})]\}_{\boldsymbol{x}\in X_{1:n}\cup\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_q\}}$ from the posterior of $[\boldsymbol{f}(\boldsymbol{x}\diamond\boldsymbol{\xi})]_{\boldsymbol{x}\in X_{1:n}\cup\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_q\},\boldsymbol{\xi}\in\Xi}$ via a simple matrix-matrix product, and use that to draw the posterior samples. This avoids inverting m $(n+q)n_{\boldsymbol{\xi}}\times(n+q)|\Xi|$ matrices, and reduces the cost of posterior sampling from $\mathcal{O}(m(n+q)^3n_{\boldsymbol{\xi}}^3)$ to $\mathcal{O}(m(n+q)^3)$. However, this computational technique cannot be used if there are black-box constraints. We refer to this method as Exp-qNParEGO.

3.F.2 Optimization Expectation Objectives with qNEHVI

The other acquisition function we consider is the qNEHVI:

$$\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}_{\boldsymbol{f}\sim P(\boldsymbol{f}|\mathcal{D})}\left[\mathrm{HVI}(\boldsymbol{f}(\boldsymbol{x})|\mathcal{P}_n)\right], \tag{3.4}$$

where $\mathcal{P}_n$ is the PF over $\boldsymbol{f}(\boldsymbol{x})_{\boldsymbol{x}'\in X_{1:n}}$. The extension of qNEHVI to the expectation objectives follows a similar path to that of qNParEGO. We replace the hypervolume improvement of $\boldsymbol{f}(\boldsymbol{x})$ in (3.4) with the hypervolume improvement of $\mathbb{E}_{\boldsymbol{\xi}\sim P(\boldsymbol{\xi})}[\boldsymbol{f}(\boldsymbol{x})]$ with respect to the PF over $\{\mathbb{E}_{\boldsymbol{\xi}\sim P(\boldsymbol{\xi})}[\boldsymbol{f}(\boldsymbol{x}')]\}_{\boldsymbol{x}'\in X_{1:n}}$. To do so, we replace the posterior samples of $\boldsymbol{f}(\boldsymbol{x})$ and $\{\boldsymbol{f}(\boldsymbol{x}')]\}_{\boldsymbol{x}'\in X_{1:n}}$ that are used in qNEHVI calculations with the posterior samples of the expectation over $P(\boldsymbol{\xi})$, which can be obtained in the same manner described above for qNParEGO. However, qNEHVI with expectation objective is often prohibitively slow because typically relatively few points from $X_{1:n}$ can be pruned as discussed in Appendix 3.E. Hence, we only evaluate a single sample approximation using RFFs, analogous to the RFF approximation of MVaR-NEHVI, which we refer to as Exp-NEHVI-RFF.

3.F.3 Challenges of Using Expectation with Feasibility-Weighted Objectives

Independently computing the expectation of the objectives and the feasibility and taking the product of the expectations, would ignore the fact that the objective functions and constraint functions are evaluated on the same perturbed designs. To account for the perturbed inputs jointly across in the objectives and constraints, we use feasibility weighted objectives. Feasibility weighting requires penalizing designs that are infeasible such that the feasibility-weighted objectives for an infeasible design are worse than the objectives for any feasible design.

Feasibility weighting can make the expectation sensitive to the range of the objectives. When evaluating a solution near the border of the feasible domain, we end up with a subset of the perturbed solutions evaluating to zero due to infeasibility and others evaluating to their respective objective values. To see how this can affect the performance, consider the following examples. Suppose that half of the perturbed solutions are infeasible and the objective values are bounded in $[0, 1]$. In this case, the feasibility weighted objective take values in $[0, 0.5]$, where it will be inferior to some other solutions due to the potential for it to be infeasible. Now, suppose that the objective values are bounded in $[100, 101]$. The infeasibility in this case will bring the feasibility weighted objective to the range of $[50, 50.5]$, which is strictly worse than any solution that is more feasible, even if by only a small fraction. If we instead set the infeasible solutions to 100 rather than zero, this would lead to the feasibility weighted objective value to $[100, 100.5]$. Note that setting the infeasible solutions to 100 is equivalent to normalizing the objectives to $[0, 1]$ before applying the feasibility and using zero for the infeasible objectives, which in theory should have no effect in the optimization performance. In practice, we typically do not know precise bounds on the objectives, and instead standardize / normalize the objectives during optimization using bounds derived on the go.

The example above highlights the effect of the range of each objective. Often, an infeasibility cost λ is used to penalize for infeasible designs to ensure that infeasible points are worse than any feasible point (for example if the objectives can take

negative values) by setting the feasibility-weighted objectives to $(\boldsymbol{f}(\boldsymbol{x}) + \lambda)\mathbb{1}[\boldsymbol{c} > \boldsymbol{0}] - \lambda$ for some $\lambda \geq 0$. However, the feasibility weighted expectation is typically sensitive to the infeasibility cost, and feasibility weighted objectives give higher value to conservative solutions if the Pareto front lies near the border of the feasible domain. This makes it difficult to determine apriori how conservatively expectation methods will act when using feasibility weighted objectives. In contrast, MVaR avoids this issue by providing high probability guarantees on the value of the feasibility-weighted objectives under input noise. For any design that is feasible with probability α, the infeasibility cost is in the tail of the multivariate CDF and has no effect on the elements of the MVaR set.

3.G Experiment Details

3.G.1 Method Details

We evaluate the following BO methods:

Methods that optimize the nominal objectives (see Daulton et al. [2021]) for details): qNParEGO, qNEHVI, and NEHVI-RFF (referred to as qNEHVI-1 in Daulton et al. [2021]), which approximates the expectation in qNEHVI with a single approximate GP sample using RFFs.

Methods that optimize the expectation objectives (Appendix 3.F): Exp-qNParEGO, and Exp-NEHVI-RFF.

Methods that optimize MVaR: MARS-NEI (Section 3.6.2), MARS-TS (Appendix 3.B), MARS-UCB (Appendix 3.B), MVaR-NEHVI (Appendix 3.D), and MVaR-NEHVI-RFF (Appendix 3.D).

We implemented all methods using the BoTorch library [Balandat et al., 2020] (except for NSGA-II), leveraging the existing implementations of NEI and qNEHVI available at `https://github.com/pytorch/botorch`. We used the implementation of NSGA-II in the PyMOO library [Blank and Deb, 2020], which is available at `https://github.com/anyoptimization/pymoo`.

For all model-based methods, we model each objective and constraint with an independent GP with a Matérn-$\frac{5}{2}$ ARD kernel [Rasmussen, 2004].[12] For methods that use scalarizations, we use composite objectives [Astudillo and Frazier, 2019]. We use maximum a posteriori estimates of the GP hyperparameters using the default priors in BoTorch. For all MC-based acquisition functions, we use $N_{MC} = 256$ QMC samples from the GP posterior. We use sample-average approximation [Balandat et al., 2020] by using fixed Quasi-MC samples from $P(\boldsymbol{\xi})$ (for robust methods) and fixed Quasi-MC base samples for all methods to approximate the expectation over the GP posterior.[13][14] This results in an approximation of the acquisition function that is a deterministic function of the input $\boldsymbol{x}$. For RFF-based methods, the approximate GP sample (using 512 random features) is also a deterministic function, which, coupled with fixed samples from $P(\boldsymbol{\xi})$, results in a deterministic approximation of the acquisition functions. The deterministic approximations of the acquisition functions enable the use of quasi-Newton methods for optimization. We optimize all acquisition functions using multi-start optimization with L-BFGS-B [Zhu et al., 1997].

For MARS and other MVaR-based methods, we use the known MVaR reference point, which would typically be supplied by the decision maker. For qNEHVI and Exp-NEHVI-RFF, we use the heuristic from Daulton et al. [2020] to adaptively infer the the reference point during the optimization (the MVaR reference point is not suitable for the nominal and expectation objectives).

For methods involving scalarizations, the objectives are normalized before applying the scalarizations. For MARS methods, the reference point is used as the lower bound and the ideal point (i.e. the component-wise maximum of each objective, Ishibuchi et al. [2018]) across the MVaR set over the previously evaluated designs (estimated using the posterior mean) is used as the upper bound for normalization. For qNParEGO, we use the ideal and nadir points (i.e. the

[12] All methods except MARS-UCB support using multi-task GPs that model for correlations between objectives [Bonilla et al., 2008].

[13] All fixed samples are re-sampled once per BO iteration.

[14] For heteroskedastic input noise processes, we fix a set base samples and use the reparameterization trick [Kingma and Welling, 2013] to sample from from the heteroskedastic input noise process using the fixed based samples—rather than directly fixing the input noise samples as we do in the case of homoskedastic noise.

component-wise minimum objective values across the PF, Ishibuchi et al. [2018]) across the PF over the previously evaluated designs. Similarly, for Exp-qNParEGO, we use the ideal and nadir points over the PF expectation objectives (estimated using the posterior mean) over the previously evaluated designs.

For feasibility weighting the objectives, we use a sigmoid function as a differentiable approximation of the indicator function as in Balandat et al. [2020]. The infeasibility cost set to be the minimum posterior mean minus six standard deviations.

For methods that use NEI and qNEHVI with exact posterior sampling, we prune the previously evaluated designs using $N_{\text{prune}} = 2048$ samples to estimate the probability that a previously evaluated design is optimal. Additionally, we cache the Cholesky decomposition of the posterior covariance matrix over $\{\boldsymbol{f}(\boldsymbol{x}' \diamond \boldsymbol{\xi})\}_{\boldsymbol{x}' \in X_{\text{pruned}}}$ and use low-rank updates to draw joint samples over $\{\boldsymbol{f}(\boldsymbol{x}' \diamond \boldsymbol{\xi})\}_{\boldsymbol{x}' \in X_{\text{pruned}} \cup \{\boldsymbol{x}\}}$ [Osborne and of Oxford, 2010].

For batch (or asynchronous) candidate generation, we use a sequential greedy approach [Wilson et al., 2018], where one new candidate is optimized at time and the joint acquisition value of all candidates $\boldsymbol{x}_1, ..., \boldsymbol{x}_i$ is optimize to select $\boldsymbol{x}_i$. For methods relying on scalarizations, a new scalarization is sampled for each new candidate.

For NSGA-II, we used the same initial sobol starting points as for the other methods. We used a population size of 10 and adjusted the number of iterations for NSGA-II according to the evaluation budget. The PyMOO default configuration was used for all other settings. In addition to the objectives, observations of the constraints are provided to NSGA-II.

3.G.2 Problem Details

In this section, we provide descriptions of the test problems. The reference points used for all problems are provided in Table 3.G.2. For each problem, we set the reference point to be slightly worse than the nadir point (using the heuristic from Ishibuchi et al. [2011]) of the MVaR set evaluated over a large grid of design points. For the Penicillin problem, we set the reference point to exclude the region of the objective space with low values of the time objective (following Liang and

Lai [2021]) as those objective trade-offs are less appealing to decision makers due to providing negligible Penicillin yield.

Toy Problem ($d = 1,\ M = 2,\ \alpha = 0.9$): This is the toy problem that was used to highlight the concepts in Figures 3.1&3.2. The noise model is given by $P(\boldsymbol{\xi}) = \mathcal{N}(\mu = \mathbf{0}, \Sigma = 0.1I_2)$. The first objective is a mixture linear-sinusoidal function, and the second objective is modified from the well-known Levy test function. The exact expressions are given as follows. The function is evaluated on $x \in [0, 0.7]$.

$$
\begin{aligned}
f^{(1)}(x) &= 30 - 30 * (p_1(x)p_4(x) + p_2(x)(1 - p_4(x)) + p_3(x)) \\
p_1(x) &= 2.4 - 10x - 0.1x^2 \\
p_2(x) &= 2x - 0.1x^2 \\
p_3(x) &= (x - 0.5)^2 + 0.1\sin(30x) \\
p_4(x) &= 1/\left(1 + \exp\left((x - 0.2)/0.005\right)\right) \\
f^{(2)}(x) &= p_5\left((x * 0.95 + 0.03) * 20 - 10\right) \\
p_5(x) &= p_6\left(1 + (x - 1)/4\right) - 0.75 * x^2 + 9.0955 \\
p_6(x) &= \left(\sin(\pi * x)\right)^2 + (x - 1)^2(1 + 10\left(\sin(\pi * x)\right)^2)
\end{aligned}
$$

GMM ($d = 2,\ M \in \{2, 3, 4\},\ \alpha \in \{0.7, 0.8, 0.9\}$): In addition to the version presented in the main text, we consider several variations of the GMM problem using different number of objectives, different noise models, and different risk levels to analyze the effects of these factors on the optimization performance of the algorithms. For all GMM problems considered, each objective is a mixture of the probability density function of three Gaussian distributions, modified from the single objective version presented in Fröhlich et al. [2020]. We present the canonical formula of the objectives and the parameters corresponding to each objective below. In the formula, $\phi(\boldsymbol{x}; \mu, \Sigma)$ is used to denote the probability density function of the multivariate Gaussian distribution with mean μ and covariance matrix Σ. The search space for all GMM problems is $\mathcal{X} = [0, 1]^2$.

The GMM problem used in the main text involves 2 objectives and uses $\alpha = 0.9$. Additional experiments in Appendix 3.I.1 use additional independent GMMs to

increase the number of objectives to 3 and 4, and evaluate performance with different settings of $\alpha \in \{0.7, 0.8, 0.9\}$. In all experiments with 3 and 4 objective GMM, we use additive noise, where $P(\boldsymbol{\xi}) = \mathcal{N}(\mu = \mathbf{0}, \Sigma = 0.05 I_M)$. Many additional noise processes are discussed and evaluated in Appendix 3.I.5, using the same 2 objective GMM problem from the main text and $\alpha = 0.9$.

$$f^{(i)}(\boldsymbol{x}) = 2\pi \sum_{j=1}^{3} var_j^{(i)} cons_j^{(i)} \phi(\boldsymbol{x}; \mu = pos_j^{(i)}, \Sigma = var_j^{(i)} I_2)$$

$$pos_j^{(i)} = \begin{cases} j=1 & j=2 & j=3 & \\ [0.2, 0.2] & [0.8, 0.2] & [0.5, 0.7] & \text{if } i=1 \\ [0.07, 0.2] & [0.4, 0.8] & [0.85, 0.1] & \text{if } i=2 \\ [0.08, 0.21] & [0.45, 0.75] & [0.86, 0.1] & \text{if } i=3 \\ [0.09, 0.19] & [0.44, 0.72] & [0.89, 0.13] & \text{if } i=4 \end{cases}$$

$$var_j^{(i)} = \begin{cases} j=1 & j=2 & j=3 & \\ 0.04 & 0.01 & 0.01 & \text{if } i=1 \\ 0.04 & 0.01 & 0.0025 & \text{if } i=2 \\ 0.04 & 0.01 & 0.0049 & \text{if } i=3 \\ 0.0225 & 0.0049 & 0.0081 & \text{if } i=4 \end{cases}$$

$$cons_j^{(i)} = \begin{cases} j=1 & j=2 & j=3 & \\ 0.5 & 0.7 & 0.7 & \text{if } i=1 \\ 0.5 & 0.7 & 0.7 & \text{if } i=2 \\ 0.5 & 0.7 & 0.9 & \text{if } i=3 \\ 0.5 & 0.7 & 0.9 & \text{if } i=4 \end{cases}$$

Constrained Branin Currin We use the open source implementation available at `https://github.com/pytorch/botorch`. See Daulton et al. [2020] for details.

Disc Brake We use the open source implementation available at `https://github.com/ryojitanabe/reproblems`. See Tanabe and Ishibuchi [2020] for details.

Penicillin Manufacturing Problem We use the open-source implementation available at `https://github.com/HarryQL/TuRBO-Penicillin`. See Liang and Lai [2021] for details. We adapt the problem by adding independent zero-mean Gaussian input noise to each parameter. The standard deviation of the input noise distribution for each parameter is listed in Table 3.G.1.

Table 3.G.1: Standard deviation for independent zero-mean Gaussian input noise for each parameter in the Penicillin Problem (reported as a percentage of the range of each parameter).

Parameter	Noise Level
Culture Volume	3%
Biomass Concentration	3%
Temperature	0.5%
Glucose Concentration	2%
Substrate Feed Rate	1%
Substrate Feed Concentration	1%
H^+ Concentration	1%

Table 3.G.2: Reference points for negative versions (i.e. multiplying the objectives by -1 to make the goal maximization of all objectives) of all problems (except the GMM and Toy problems, which are designed for maximization).

Problem	Reference Point
Toy Problem	[-14.1951, -3.1887]
Disc Brake	[-5.89, -3.27]
Constrained Branin Currin (heteroskedastic noise)	[-194.9376, -12.2969]
Constrained Branin Currin (homoskedastic noise)	[-195.4667, -12.4984]
Penicillin	[5.657, -64.1, -340.0]
GMM ($M = 2, \alpha = 0.9$, multiplicative noise)	[0.3752, 0.3548]
GMM ($M = 2, \alpha = 0.9$, correlated noise)	[0.2727, 0.2583]
GMM ($M = 2, \alpha = 0.8$, heteroskedastic noise)	[0.3465, 0.3036]
GMM ($M = 2, \alpha = 0.9$, homoskedastic noise, $\sigma = 0.05$)	[0.2756, 0.2368]
GMM ($M = 2, \alpha = 0.9$, homoskedastic noise, $\sigma = 0.1$)	[0.1047, 0.1112]
GMM ($M = 2, \alpha = 0.9$, homoskedastic noise, $\sigma = 0.2$)	[0.0160, 0.0131]
GMM ($M = 3, \alpha = 0.9$, homoskedastic noise, $\sigma = 0.05$)	[0.2733, 0.0051, 0.1538]
GMM ($M = 3, \alpha = 0.9$, homoskedastic noise, $\sigma = 0.05$)	[0.2733, 0.0051, 0.1538]
GMM ($M = 3, \alpha = 0.8$, homoskedastic noise, $\sigma = 0.05$)	[0.0420, 0.0180, 0.1952]
GMM ($M = 3, \alpha = 0.7$, homoskedastic noise, $\sigma = 0.05$)	[0.0537, -0.0517, -0.0021]
GMM ($M = 4, \alpha = 0.9$, homoskedastic noise, $\sigma = 0.05$)	[0.0264, -0.0396, 0.0619, 0.1689]
GMM ($M = 4, \alpha = 0.8$, homoskedastic noise, $\sigma = 0.05$)	[0.0322, -0.0398, 0.1168, -0.0023]

3.G.3 Evaluation Details

The global MVaR set is unknown and is approximated by taking the union of the MVaR sets of all designs evaluated across all methods and all replications. We take this approach because even using an evolutionary algorithm to optimize MVaR is nontrivial, since MVaR maps a single design to a set of points and is relatively computationally intensive to evaluate. To evaluate the performance of a given method, we use $n_{\boldsymbol{\xi}} = 512$ (except for 4 objective GMM, where we use $n_{\boldsymbol{\xi}} = 256$) to compute a high-fidelity estimate of the MVaR set across the designs selected during optimization by the method. We similarly use the same $n_{\boldsymbol{\xi}} = 512$

samples to estimate the true MVaR set (by considering all designs evaluated across all methods and all replications).

3.H Wall Times

In Table 3.H.1, we present the time it takes to run a single BO iteration using all algorithms we considered in this paper. We include the runtimes for the four problems from the main text. Wall times for additional problems including several problems with 3 and 4 objectives are provided in Table 3.I.1 (these problems are described in Appendix 3.I.1). As we discuss in Appendix 3.D, MVaR-NEHVI, when not computationally infeasible, is quite expensive to run, making it an impractical method for most problems. Although MVaR-NEHVI-RFF provides a much cheaper and highly performant approximation, we see that it also runs into computational limitations as the size of the MVaR set grows (e.g. for Penicillin with 3 objectives); this is also pronounced in Table 3.I.1 on the problems with 3 and 4 objectives. For the MARS family of methods, we see overall quite reasonable runtimes, with the most expensive one, MARS-NEI, taking on average 41.4 seconds on the most expensive problem instance we considered. MARS-TS offers a cheaper alternative to MARS-NEI, with its average runtime remaining below 10 seconds on all experiments. As we show later in the Appendix, the performance of MARS-TS typically trails closely behind MARS-NEI, making it a strong alternative when the algorithm runtime is of the essence.

Table 3.H.1: The wall time (in seconds) per BO iteration. The experiments were timed on a shared cluster using 4 CPU cores, 1 GPU, and 16 GB of RAM. We report the mean and 2 standard errors over 20 trials. An N/A entry denotes that we did not attempt to run a particular experiment (e.g., because the method does not support the problem setting), whereas an OOM entry denotes that we attempted but the experiment did not run due to scalability limitations. The three top-performing algorithms, with respect to the final average MVaR HV regret in each experiment are highlighted using ***best***, ***second***, ***third***, respectively.

Algorithm (d, M, V, α)	GMM $(2, 2, 0, 0.9)$	Constrained BC $(2, 2, 1, 0.7)$	Disc Brake $(4, 2, 4, 0.95)$	Penicillin $(7, 3, 0, 0.8)$
Sobol	0.4 (±0.6)	0.9 (±1.0)	0.9 (±1.0)	2.9 (±1.5)
qNParEGO	2.5 (±2.3)	5.6 (±4.4)	7.9 (±15.1)	21.2 (±25.6)
qNEHVI	2.5 (±2.0)	9.8 (±8.9)	16.9 (±10.3)	23.6 (±41.0)
qNEHVI-RFF	0.8 (±0.7)	2.1 (±3.2)	2.7 (±5.4)	***second*** 5.0 (±3.9)
Exp-qNParEGO	3.3 (±2.4)	10.7 (±11.1)	23.3 (±166.3)	121.9 (±119.0)
Exp-NEHVI-RFF	0.9 (±1.0)	7.2 (±10.8)	***best*** 3.3 (±7.3)	***third*** 5.1 (±3.3)
MARS-NEI	3.9 (±3.0)	***second*** 8.4 (±5.3)	10.3 (±45.4)	***best*** 41.4 (±60.1)
MARS-TS	***third*** 3.3 (±3.6)	***third*** 3.3 (±3.7)	3.1 (±6.5)	6.7 (±5.8)
MARS-UCB	8.2 (±11.0)	N/A	N/A	12.3 (±14.1)
MVaR-NEHVI	***best*** 145.6 (±130.7)	N/A	***second*** 243.2 (±154.)	N/A
MVaR-NEHVI-RFF	***second*** 1.8 (±1.7)	***best*** 10.0 (±11.9)	***third*** 2.9 (±3.1)	OOM

3.I Additional Experiments

3.I.1 Additional Test Problems

In addition to the problems presented in the main text, we studied the performance of the algorithms on the Toy problem used for illustrations in Figures 3.1 and 3.2, and the 3 and 4 objective variations of the GMM problem. These problems are described in detail in Appendix 3.G.2. In addition to the acquisition functions presented in the main text, we ran Exp-NEHVI-RFF, MVaR-NEHVI-RFF, MARS-UCB, and MARS-TS. The results of these experiments are presented in Figure 3.I.1 for the Toy problem and Figure 3.I.2 for the GMM problems, and the runtimes of the algorithms are reported in Table 3.I.1. We see that MARS-NEI is overall the best performing method, with MARS-TS typically following closely. MARS-UCB appears to be less reliable, demonstrating significantly worse performance in most experiments. In addition, the MVaR-NEHVI-RFF is missing from all but two of the experiments, which is due to the method running into the scalability limitations discussed in Appendix 3.D.

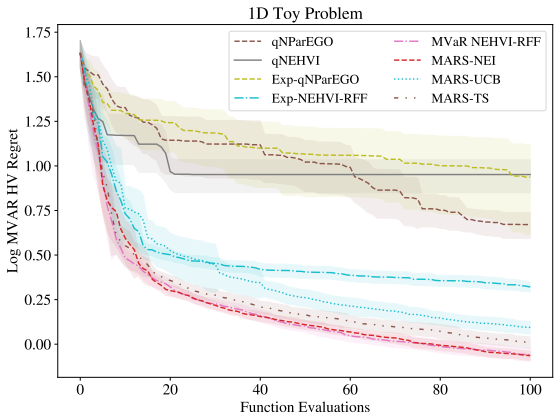

Figure 3.I.1: The log MVaR hypervolume regret on the Toy problem. We plot means and 2 standard errors across 20 trials.

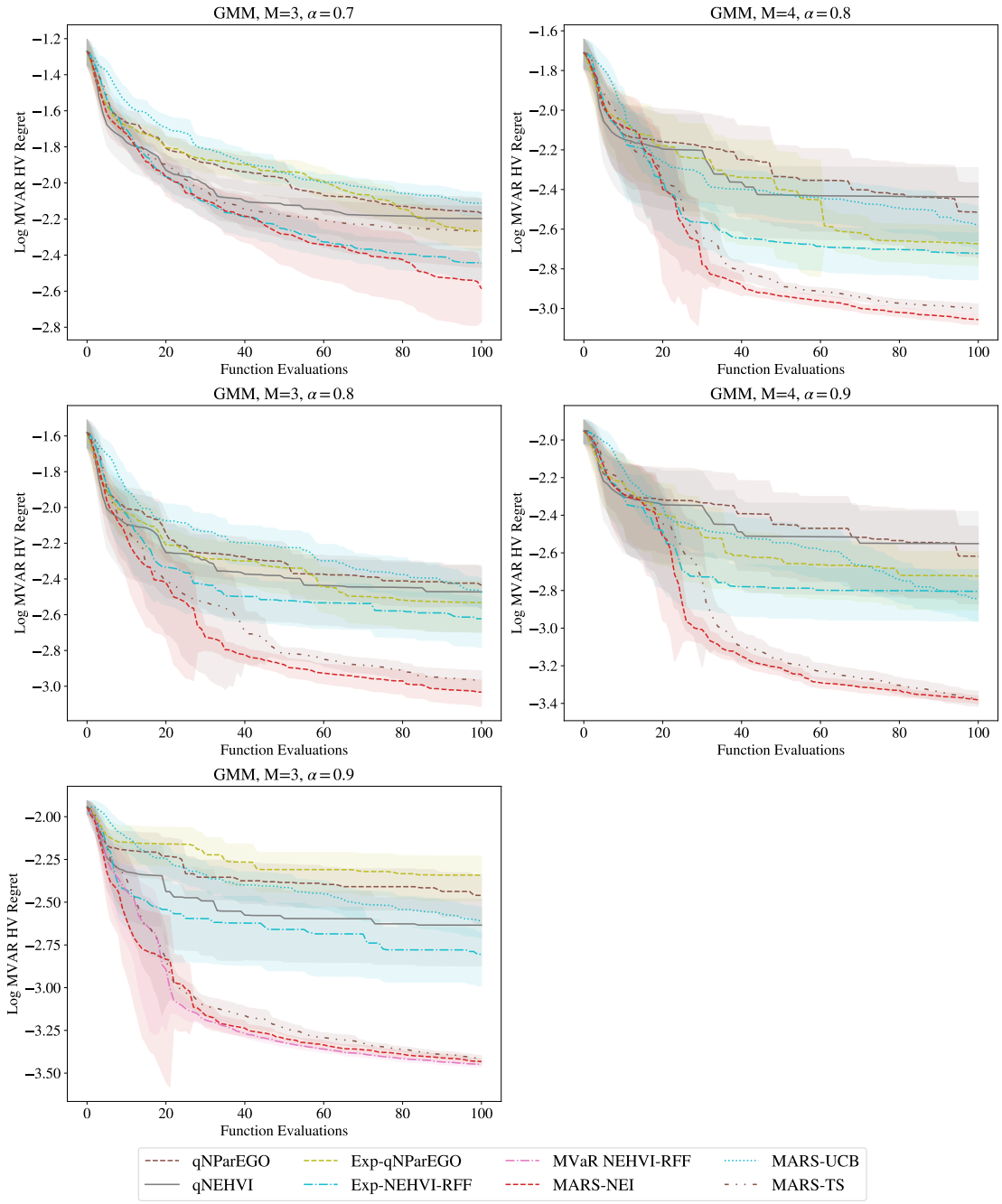

Figure 3.I.2: The log MVaR hypervolume regret on 3 and 4-Objective GMM problems. We plot means and 2 standard errors across 20 trials.

Table 3.I.1: The wall time (in seconds) per BO iteration for the additional problems. The experiments were timed on a shared cluster using 4 CPU cores, 1 GPU, and 16 GB of RAM. We report the mean and 2 standard errors over 20 trials. An OOM entry denotes that the experiment did not run due to scalability limitations. The three top-performing algorithms, with respect to the final average MVaR HV regret in each experiment are highlighted using ***best***, ***second***, ***third***, respectively.

Algorithm	Toy Problem	GMM, M=3			GMM, M=4	
(d, M, α)	$(1, 2, 0.9)$	$(2, 3, 0.7)$	$(2, 3, 0.8)$	$(2, 3, 0.9)$	$(2, 4, 0.8)$	$(2, 4, 0.9)$
Sobol	0.4 (±0.3)	1.9 (±2.2)	1.2 (±1.4)	0.5 (±0.3)	3.3 (±3.3)	0.8 (±0.2)
qNParEGO	2.2 (±2.0)	4.5 (±2.6)	3.4 (±2.6)	1.4 (±1.5)	8.9 (±10.2)	4.5 (±12.9)
qNEHVI	3.0 (±2.5)	23.3 (±26.4)	20.2 (±22.8)	23.8 (±43.8)	57.3 (±75.4)	48.0 (±55.2)
qNEHVI-RFF	0.4 (±0.3)	***third*** 3.1 (±5.8)	1.7 (±1.0)	0.7 (±0.5)	10.0 (±91.8)	2.1 (±4.0)
Exp-qNParEGO	13.1 (±23.5)	8.0 (±12.8)	6.2 (±6.4)	13.4 (±54.2)	10.6 (±10.6)	8.1 (±18.3)
Exp-NEHVI-RFF	0.4 (±0.3)	***second*** 4.0 (±10.3)	***third*** 2.1 (±6.3)	0.8 (±0.8)	***third*** 5.3 (±4.0)	1.9 (±1.5)
MARS-NEI	***best*** 10.4 (±21.1)	***best*** 11.1 (±14.4)	***best*** 10.7 (±23.0)	***second*** 8.8 (±13.2)	***best*** 27.5 (±221.1)	***best*** 13.5 (±15.9)
MARS-TS	***third*** 0.9 (±0.7)	5.7 (±8.9)	***second*** 3.8 (±2.9)	***third*** 1.8 (±2.5)	***second*** 9.9 (±27.1)	***second*** 4.2 (±4.2)
MARS-UCB	6.7 (±7.4)	14.4 (±22.5)	10.4 (±11.1)	10.3 (±12.3)	17.6 (±19.1)	***third*** 12.1 (±11.9)
MVaR-NEHVI-RFF	***second*** 2.6 (±2.2)	OOM	OOM	***best*** 3.0 (±2.1)	OOM	OOM

3.I.2 Comparison with qNEHVI Based Methods

Table 3.I.2: The final MVaR HV regret obtained using each method. We report the mean and 2 standard errors over 20 trials. An N/A entry denotes that we did not attempt to run a particular experiment (e.g., because the method does not support the problem setting), whereas an OOM entry denotes that we attempted but the experiment did not run due to scalability limitations. The three top-performing algorithms, with respect to the final average MVaR HV regret in each experiment are highlighted using ***best***, ***second***, ***third***, respectively.

Algorithm	GMM	Constrained BC	Disc Brake	Penicillin	Toy Problem	GMM, M=4
(d, M, V, α)	$(2, 2, 0, 0.9)$	$(2, 2, 1, 0.7)$	$(4, 2, 4, 0.95)$	$(7, 3, 0, 0.8)$	$(1, 2, 0, 0.9)$	$(2, 4, 0, 0.8)$
Scale	1×10^{-4}	1×10^{1}	1×10^{0}	1×10^{4}	1×10^{0}	1×10^{-3}
Sobol	65.37 (±10.23)	3.12 (±0.36)	19.39 (±1.35)	1.68 (±0.05)	2.38 (±0.27)	7.39 (±1.35)
qNParEGO	21.54 (±7.68)	3.28 (±0.41)	19.10 (±1.83)	1.73 (±0.09)	4.69 (±0.78)	3.04 (±1.24)
qNEHVI	31.21 (±14.34)	5.16 (±0.76)	12.33 (±1.77)	1.65 (±0.12)	8.95 (±1.87)	3.66 (±1.47)
qNEHVI-RFF	30.54 (±12.13)	5.60 (±0.81)	17.11 (±1.97)	***second*** 1.38 (±0.09)	5.97 (±0.77)	2.18 (±0.62)
Exp-qNParEGO	10.29 (±2.92)	2.73 (±0.77)	3.43 (±0.57)	1.52 (±0.10)	8.65 (±4.53)	2.12 (±0.47)
Exp-NEHVI-RFF	13.83 (±5.95)	1.23 (±0.05)	***best*** 1.02 (±0.05)	***third*** 1.46 (±0.06)	2.09 (±0.13)	***third*** 1.90 (±0.50)
MARS-NEI	1.45 (±0.12)	***second*** 0.70 (±0.04)	2.96 (±0.13)	***best*** 1.06 (±0.09)	***best*** 0.86 (±0.05)	***best*** 0.88 (±0.05)
MARS-TS	***third*** 0.92 (±0.07)	***third*** 0.85 (±0.05)	3.20 (±0.21)	1.49 (±0.05)	***third*** 1.02 (±0.08)	***second*** 1.00 (±0.06)
MARS-UCB	26.69 (±3.28)	N/A	N/A	1.77 (±0.01)	1.24 (±0.10)	2.61 (±0.64)
MVaR-NEHVI	***best*** 0.57 (±0.01)	N/A	***second*** 1.02 (±0.04)	N/A	1.76 (±0.15)	N/A
MVaR-NEHVI-RFF	***second*** 0.64 (±0.03)	***best*** 0.57 (±0.02)	***third*** 1.92 (±0.06)	OOM	***second*** 0.87 (±0.07)	OOM

As discussed in Section 3.6 and detailed in Appendix 3.D, MVaR can be optimized using qNEHVI based methods. However, this comes with serious computational challenges, some of which are eased if we use qNEHVI with RFF draws. In this section, we present results comparing MARS-NEI with Exp-NEHVI-RFF and MVaR-NEHVI-RFF using the test problems from the main

text. MVaR-NEHVI was also ran on two of the problems to provide a point of reference for its performance. In addition, we present both the standard qNEHVI, which uses GPs, as well as its RFF counterpart as a reference point on how the use of RFFs affects the performance of the methods on a given problem.

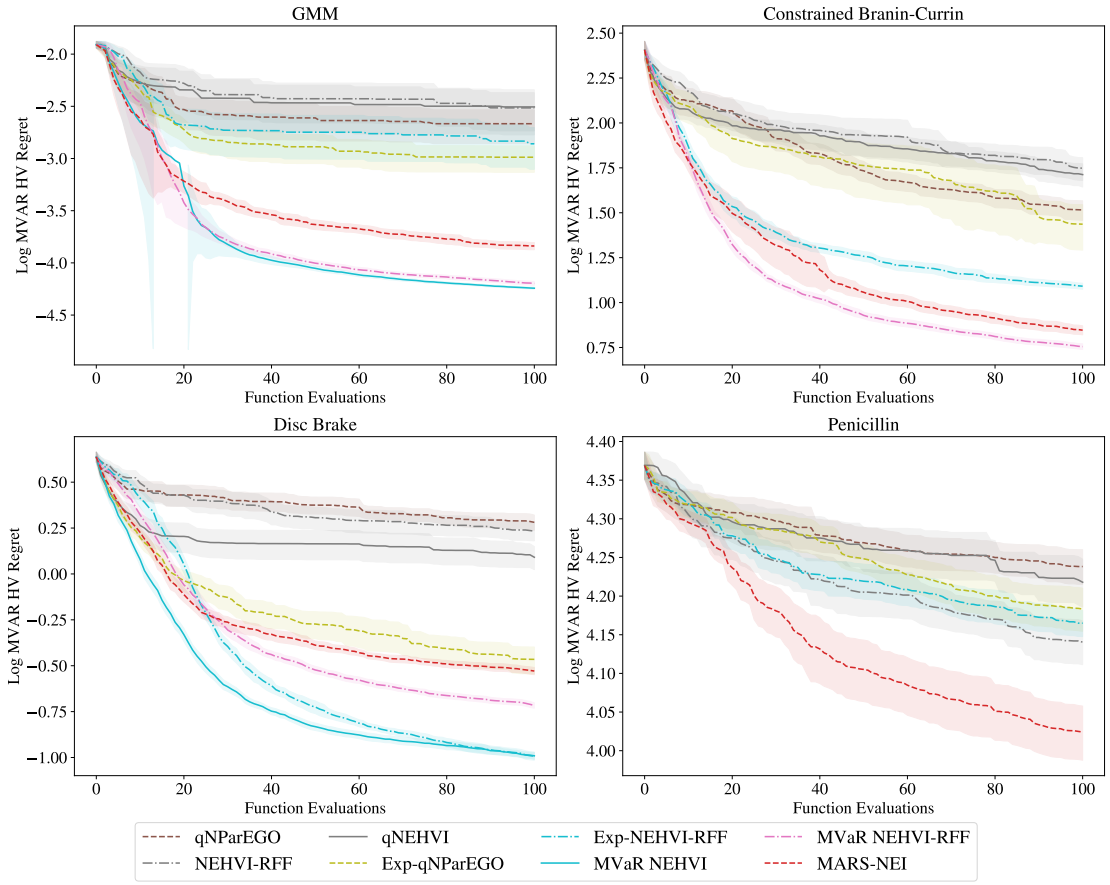

Figure 3.I.3: A comparison of the methods from evaluated in the main text with additional qNEHVI-based methods: NEHVI-RFF, Exp-NEHVI-RFF, MVaR-NEHVI, and MVaR-NEHVI-RFF.

Figure 3.I.3 and Table 3.I.2 show that although there are instances where qNEHVI methods outperform MARS-NEI, MARS-NEI remains competitive throughout. We see that qNEHVI and its RFF counterpart are competitive, without a clear winner across problems. The expectation qNEHVI outperforms the expectation qNParEGO in all but one problem, highlighting the benefit of using a method that aims to directly maximize the hypervolume in an MO setting. Lastly, it is worth highlighting that the MVaR-NEHVI-RFF is not

reported for the Penicillin problem, which is due to the scalability limitations of IEP (discussed in Appendix 3.D), preventing the hypervolume improvement computations from running with the available GPU memory. See Appendix 3.I.1 for additional experiments comparing qNEHVI based methods.

3.I.3 Comparison of Methods Optimizing MVaR

In Figure 3.I.4 and Table 3.I.2, we present results from the test problems from the main text showing the performance of all acquisition functions that we proposed for optimizing MVaR. Results for additional problems comparing are presented in Appendix 3.I.1. We observe that MVaR-NEHVI-RFF is a reasonably cheap method (see Table 3.H.1 for runtimes) that performs quite well on smaller problem instances. However, in larger problem instances where the size of the MVaR set is large (> 10) it starts running into scalability limitations and no-longer works. Among the class of MARS methods, we find that MARS-NEI consistently performs quite well, with MARS-TS being a close second and a slightly cheaper alternative. On the other hand, MARS-UCB proves to be not as reliable, performing worse than non-robust methods in some problems. We attribute this to its fundamental reliance on the parameter $\zeta_{n+1}^{(i)}$, which would have to be tuned on a problem by problem basis to optimize its performance. In light of all the results, we recommend MARS-NEI as a broadly applicable and high performing method for optimizing MVaR, and recommend MVaR-NEHVI-RFF as an alternative when the size of the MVaR set is small.

3.I.4 Parallel Evaluations

In Table 3.I.3, we present results demonstrating the effect of varying batch size on the optimization performance of MARS-NEI and MVaR-NEHVI-RFF. As expected, the results show that the performance of both algorithms degrade as the batch size increases. Interestingly, the degradation is rather minimal for MARS-NEI, while the effects of increasing batch size seem to be rather significant for MVaR-NEHVI-RFF. We see that MVaR-NEHVI-RFF underperforms all versions of MARS-NEI

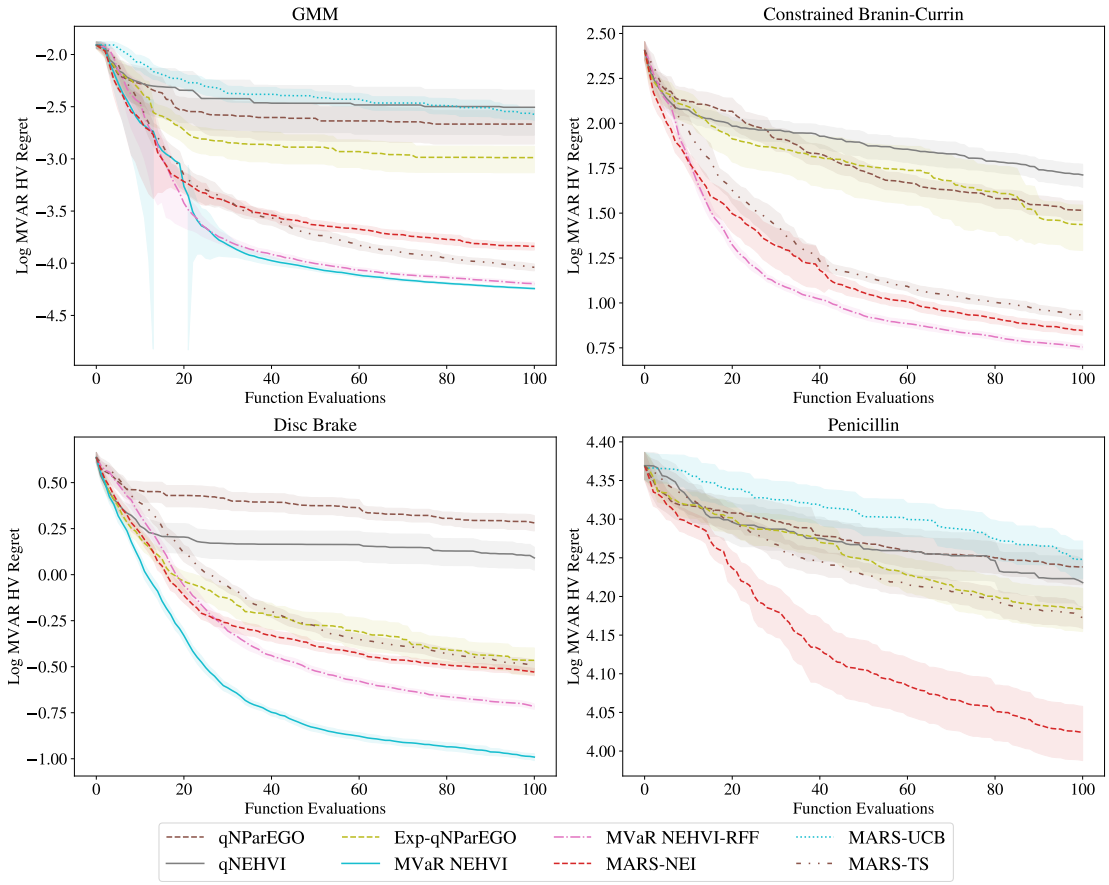

Figure 3.I.4: A comparison of the methods from evaluated in the main text with additional methods for optimizing MVaR: MARS-TS, MARS-UCB, MVaR-NEHVI-RFF, and MVaR-NEHVI.

even with a batch size of 2. In addition, there are fewer results presented for MVaR-NEHVI-RFF, which is due to the hypervolume improvement computations using IEP running into scalability limits (recall that IEP scales exponentially in the size of the joint MVaR set of the current batch of candidates). These results make a strong case for using MARS-NEI whenever one wished to evaluate candidates in parallel.

3.I.5 Effect of Noise Level

The location of robust designs on a problem depends on many factors, including the magnitude of the input noise. In the edge case where there is no input noise present, the robust designs will be the same as the nominal Pareto optimal designs. As the magnitude of the input noise increases, the robust designs may start to deviate from the nominally optimal designs, with the exact behavior typically being unpredictable

Table 3.I.3: Effect of the batch size on optimization performance. We report the final MVaR HV regret and 2 standard errors from 20 trials. OOM denotes that the method ran into scalability issues and did not run.

Algorithm	1D Toy Problem ($d = 1, M = 2$)	Constrained Branin-Currin ($d = 2, M = 2, V = 1$)
MARS-NEI, $q = 1$	0.86 ($\pm$0.05)	5.43 ($\pm$0.27)
MARS-NEI, $q = 2$	0.92 ($\pm$0.04)	5.20 ($\pm$0.21)
MARS-NEI, $q = 4$	0.92 ($\pm$0.07)	5.58 ($\pm$0.38)
MARS-NEI, $q = 8$	0.93 ($\pm$0.08)	5.59 ($\pm$0.41)
MVaR qNEHVI RFF, $q = 1$	0.87 ($\pm$0.07)	4.36 ($\pm$0.13)
MVaR qNEHVI RFF, $q = 2$	1.03 ($\pm$0.05)	6.16 ($\pm$0.36)
MVaR qNEHVI RFF, $q = 4$	1.21 ($\pm$0.07)	OOM
MVaR qNEHVI RFF, $q = 8$	OOM	OOM

and heavily dependent on the problem and noise structure. In Figure 3.I.5, we present results on the GMM problem from the main text under various noise models, demonstrating how the performance of the algorithms change in response to changes in the noise model. The list of noise models considered in this study are as follows:

- Homoscedastic normal noise, std = 0.05: $P(\boldsymbol{\xi}) = \mathcal{N}(\mu = \mathbf{0}, \Sigma = 0.05 I_2)$.
- Homoscedastic normal noise, std = 0.10: $P(\boldsymbol{\xi}) = \mathcal{N}(\mu = \mathbf{0}, \Sigma = 0.10 I_2)$.
- Homoscedastic normal noise, std = 0.20: $P(\boldsymbol{\xi}) = \mathcal{N}(\mu = \mathbf{0}, \Sigma = 0.20 I_2)$.
- Heteroscedastic normal noise, std = 0.2X: $P(\boldsymbol{\xi}; \boldsymbol{x}) = \mathcal{N}(\mu = \mathbf{0}, \Sigma = 0.2S)$ with $S = [x_1, 0; 0, x_2]$ is the 2×2 matrix with the given entries.
- Correlated normal noise: $P(\boldsymbol{\xi}) = \mathcal{N}(\mu = \mathbf{0}, \Sigma = 0.001S)$ with $S = [2.5, -2; -2, 2.5]$.
- Multiplicative noise model from the main text: $\boldsymbol{x} \diamond \boldsymbol{\xi} := \boldsymbol{x}\boldsymbol{\xi}'$, where $\boldsymbol{\xi}' \sim \mathcal{N}(\mu = \mathbf{1}, \Sigma = 0.07 I_2)$.

We see that MARS-NEI consistently outperforms the alternatives, except for the under homoskedastic noise with a large standard deviation of 0.2. In this setting, the Exp-qNParEGO is slightly ahead, which is not too surprising since the expectation and MVaR optimal designs happen to be in the same part of the solution space under this noise model.

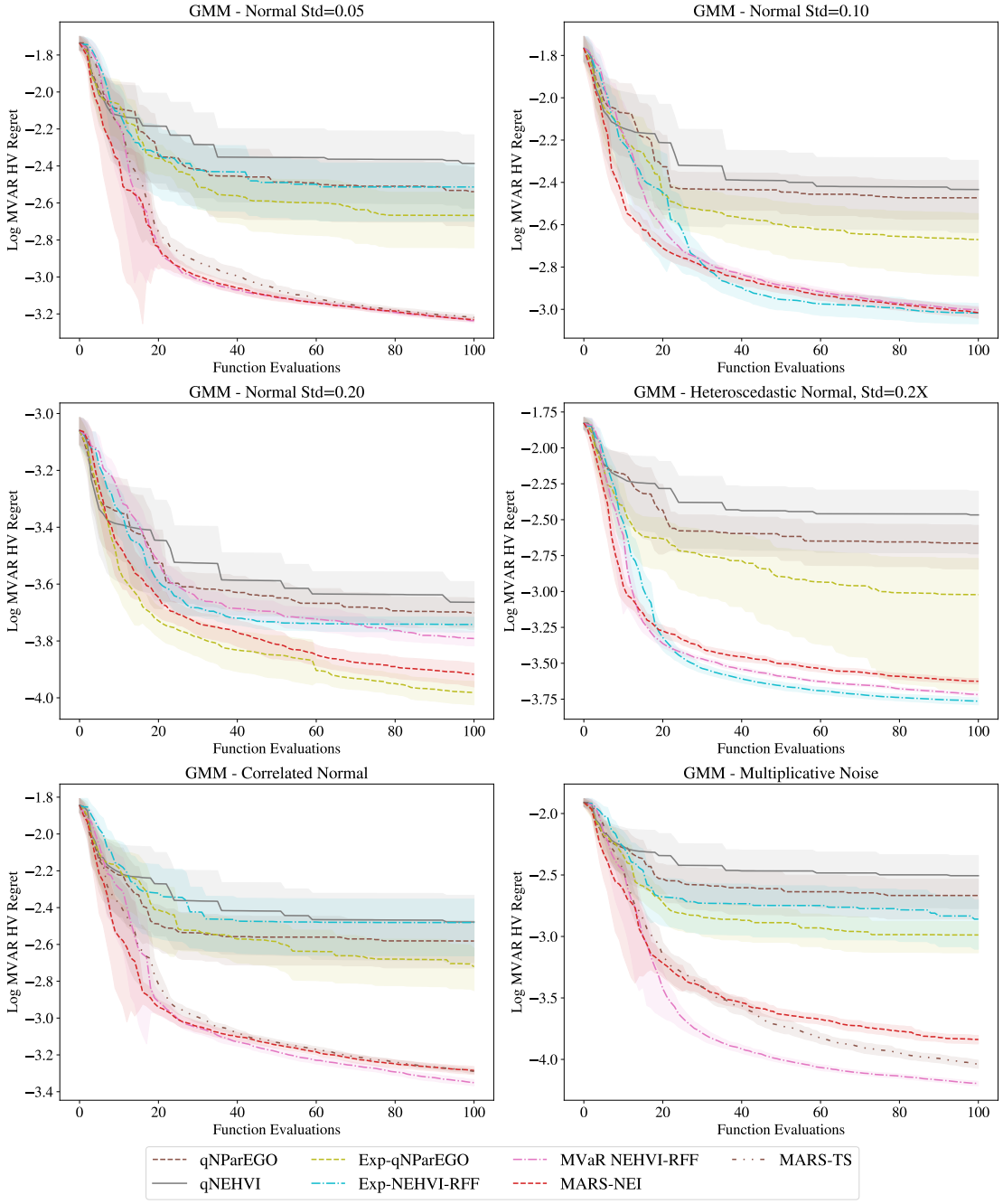

Figure 3.I.5: The effect of different noise models on the GMM problem.

In addition to the GMM problem, the Constrained Branin Currin problem was also ran under multiple noise models. Along with the heteroscedastic noise model used in the main text, we also studied it using a simple homoscedastic noise model: $P(\boldsymbol{\xi}) = \mathcal{N}(\mu = \mathbf{0}, \Sigma = 0.05 I_2)$. The plots for these are shown in Figure 3.I.6, demonstrating that MARS performs consistently under both noise models.

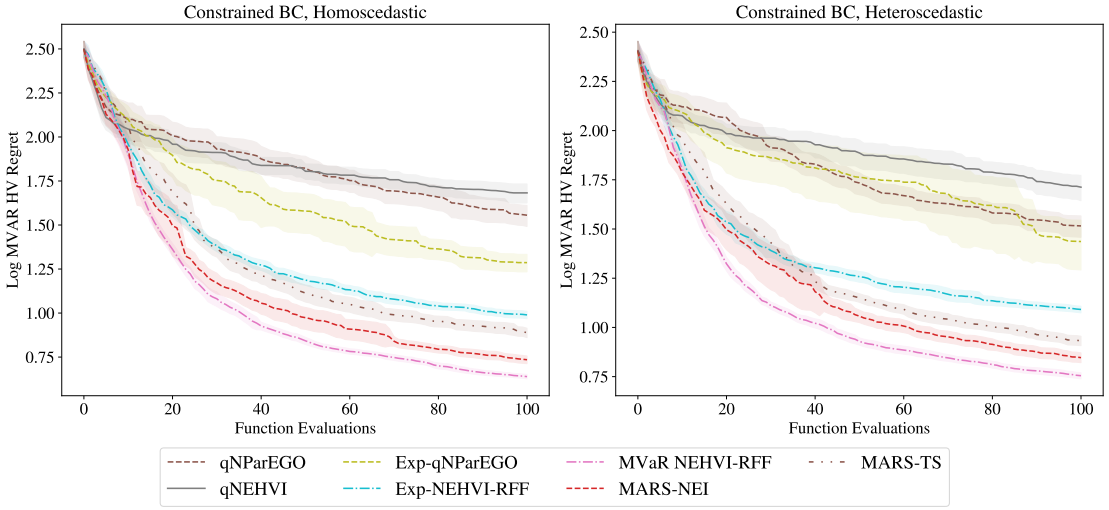

Figure 3.I.6: The effect of different noise models on the Constrained Branin Currin problem.

3.I.6 Effect of the Number of Samples on Optimization Performance

In a final side study, we analyze the effect of varying $n_{\boldsymbol{\xi}}$ on the optimization performance of the acquisition functions optimizing expectation and MVaR. We attempted to run the Constrained Branin Currin and Disc Brake experiments with $n_{\boldsymbol{\xi}} \in \{8, 16, 32, 64, 96, 128\}$ and included the results of the algorithms that successfully completed without running into scalability issues, such as getting an out-of-memory error.

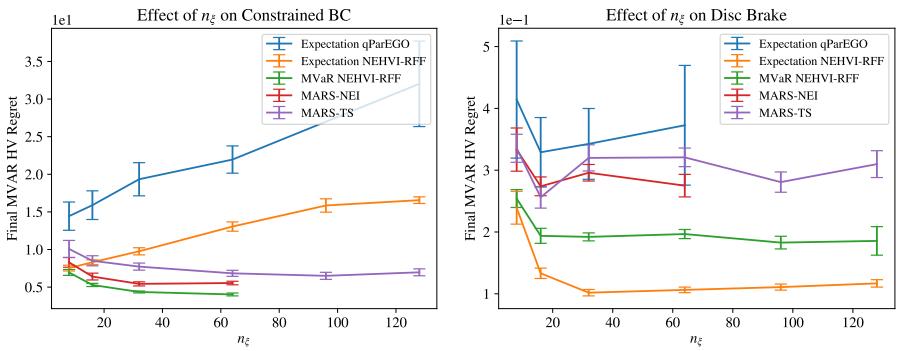

Figure 3.I.7: Final MVaR hypervolume regret obtained using different $n_{\boldsymbol{\xi}}$ on Constrained Branin-Currin and Disc Brake.

The results are shown in Figure 3.I.7. We see that the methods using GPs (Exp-qNParEGO and MARS-NEI), do not scale beyond $n_{\boldsymbol{\xi}} = 64$ due to the cubic

complexity (with respect to the number of points and $n_{\boldsymbol{\xi}}$) of posterior sampling (see Appendix 3.D.2, the remaining methods avoid this issue since the RFF draws are deterministic functions). Overall, the results for MVaR methods show that too small of an $n_{\boldsymbol{\xi}}$ leads to poor performance, while increasing it much beyond our default value of $n_{\boldsymbol{\xi}} = 32$ does not yield any a significant benefit, at least in these problems. On Constrained Branin-Currin, we observe that the performance of the expectation methods degrade as $n_{\boldsymbol{\xi}}$ increases, which we attribute to these methods becoming more proficient at differentiating the expectation and MVaR optimal regions (which are not co-located), thus focusing their sampling away from the MVaR optimal region.

3.J Efficient Methods for Computing MVaR

Before going into the discussion, we note that the this section presents the MVaR computation for a random variable to be minimized. This simplifies the discussion by enabling the use of common terms such as CDF and quantile, since the MVaR, as originally defined in Prèkopa [2012], corresponds to the α quantile of a random variable to be minimized. In the maximization setting studied in this paper, the MVaR, as defined in Definition 3.5.2, can be computed by first computing the MVaR of the negative of the random variable, as discussed here, then negating the result.

The existing algorithms for computing the MVaR (e.g., those presented by Prèkopa [2012]) presume the availability of a cheap to evaluate CDF of the random variable of interest. In the general setting we consider, with $\boldsymbol{f}$ being an arbitrary function, such as a sample path of the GP, the random variable $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$ (induced by $\boldsymbol{\xi} \sim P(\boldsymbol{\xi})$) does not admit a known CDF. Thus, to compute a QMC estimate of MVaR, we first need to compute the empirical CDF corresponding to the QMC samples of $\boldsymbol{f}(\boldsymbol{x} \diamond \boldsymbol{\xi})$.

Computing the empirical CDF of a random variable is a conceptually simple operation. All we need to do is to count the number of samples that are dominated by a given point and divide that by the total number of samples. Keeping all

other objectives constant, the empirical CDF can be seen as a step function over the domain of the given objective that changes its value only at the points that correspond to one of the sample values of that objective. Since there are $n_{\boldsymbol{\xi}}$ samples, ignoring the possibility that some samples may have equal value for some objectives, this defines an M-dimensional grid with $n_{\boldsymbol{\xi}}$ points on each dimension, on which the empirical CDF can change its value. Thus, to fully compute the empirical CDF, we need to compute the number of samples that dominate a grid of $n_{\boldsymbol{\xi}}^M$ points, at a cost of $\mathcal{O}(M)$ comparison per point, leading to a total $\mathcal{O}(n_{\boldsymbol{\xi}}^M M)$ cost for computing the empirical CDF.

Once the empirical CDF is computed, the MVaR set can easily be computed by taking the Pareto set of the points on the grid with a CDF greater than or equal to α. This part of the computation, fortunately, has a lower complexity than the CDF computations.

A careful reader might have noticed that the MVaR (in the minimization setting) is bounded from below by the independent VaR of each objective and bounded from above by the maximum value observed for that given objective. We can leverage this fact to lower the cost of computing MVaR significantly. Instead of considering the full grid of $n_{\boldsymbol{\xi}}^M$ points, we can only compute the empirical CDF for the grid formed by the objective values that exceed the independent VaR of each objective, of which there are $(1-\alpha)n_{\boldsymbol{\xi}}$, reducing the complexity of MVaR computations to $\mathcal{O}((1-\alpha)^M n_{\boldsymbol{\xi}}^M M)$.

Within our code base, we provide two implementations for computing MVaR. One implementation is geared towards batched calculations with small to moderate $n_{\boldsymbol{\xi}}$, and the other is geared towards less memory intensive calculations with large $n_{\boldsymbol{\xi}}$. Both utilize efficient vectorized computations to exploit modern computing hardware. However, even with these highly optimized implementations, computing MVaR can easily become a bottleneck in a BO method, since MVaR has to be computed many times during acquisition optimization. Thus, the methods for direct optimization of MVaR that we present can still be prohibitively expensive unless the objective evaluations are significantly expensive.

3.K Multivariate Extensions of CVaR

CVaR is another popular risk measure that is commonly used with univariate random variables [Rockafellar and Uryasev, 2002]. Similar to VaR, it has also been extended to the multi-variate case by various authors (e.g., Cousin and Di Bernardino [2014] and Meraklı and Küçükyavuz [2018]). However, these extensions in general do not admit a natural interpretation, whereas MVaR provides interpretable objective specifications that the objectives (under input noise) for a given design will meet with high-probability.

Endnote

Goal in Robust Multi-Objective Optimization

In this work, we formulated the problem as identifying the global MVaR set, which is akin to a union of robust Pareto frontiers. However, this is not the only way to formulate the problem of robust multi-objective optimization. For example, if the decision-maker knows their preferences in advance and can specify those preferences mathematically, then it could be advantageous to directly incorporate that prior knowledge. For example, if a decision maker can express their preferences as a linear combination of the objectives, then optimizing the VaR of that linear scalarization is a more targeted robust optimization approach. Consider the case where there are two objectives and $\boldsymbol{f}(\boldsymbol{x}_1 \diamond \boldsymbol{\xi}) = (0, 0)$ with probability 0.5 and $\boldsymbol{f}(\boldsymbol{x}_1 \diamond \boldsymbol{\xi}) = (1, 0)$ with probability 0.5. Let $\boldsymbol{f}(\boldsymbol{x}_2 \diamond \boldsymbol{\xi}) = (0, 0)$ with probability 0.5 and $\boldsymbol{f}(\boldsymbol{x}_2 \diamond \boldsymbol{\xi}) = (0, 1)$ with probability 0.5. Then, given a reference point dominated by (0,0), the MVaR sets of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have equivalent hypervolumes and neither design is strictly better than the other in the absence of preferences. This problem is true of multi-objective optimization in general: if preferences are known, typically it will be advantageous to exploit them.

In the case where the decision maker's preferences are not known a priori, there are alternative ways of combining the MVaR sets of multiple designs beyond the union operator considered in this work. For example, convex-mixing of strategies and stochastic strategy sampling could be interesting avenues for further work.

Parameters in Real World Problems

For the real world problems, we chose α to be a reasonable value that was not too close to 1. As α approaches 1 or 0, the VaR becomes challenging to approximate

via Monte Carlo due to its dependence on low-probability events. The noise levels for the Disc Brake problem come from Emch and Parkinson [1994], since we are not domain experts in this application. For the penicillin problem, we chose the input noise levels in order to create the problem for benchmarking purposes.

Plots

Unless specified otherwise, all plots (including those in the appendix) report and mean and 2 standard errors of the mean over 20 replications.

Errata

In Section 3.6, the statement "The shaded region in right plot in Figure 3.2 illustrates the region that dominates $\boldsymbol{z}$" should read "The shaded region in the right plot in Figure 3.2 illustrates the region that dominates $\boldsymbol{z}$".

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Robust Multi-Objective Bayesian Optimization Under Input Noise
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Samuel Daulton, Sait Cakmak, Maximilian Balandat, Michael A. Osborne, Enlu Zhou, and Eytan Bakshy. Robust multi-objective Bayesian optimization under input noise. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4831–4866. PMLR, 17– 23 Jul 2022. URL https://proceedings.mlr. press/v162/daulton22a.html.

Student Confirmation

Student Name:	Samuel Daulton		
Contribution to the Paper	I thought of and derived novel theoretical results that naturally motivate several methods for addressing the problem of robust multi-objective Bayesian optimization. I have implemented these methods as well as relevant baselines, identified and devised suitable benchmark problems. Together with my collaborator Sait at Georgia Tech, we empirically validated the performance of the method.		
Signature	Samuel James Daulton	Date	28 February 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Michael Osborne, Professor of Machine Learning		
Supervisor comments	I endorse the description above, which I understand to be correct. Sam indisputably made a substantial contribution to the publication.		
Signature		Date	28 February 2023

4

Multi-Objective Bayesian Optimization over High-Dimensional Search Spaces

Contents

4.1 Abstract

Many real world scientific and industrial applications require optimizing multiple competing black-box objectives. When the objectives are expensive-to-evaluate, multi-objective Bayesian optimization (BO) is a popular approach because of its high sample efficiency. However, even with recent methodological advances, most existing multi-objective BO methods perform poorly on search spaces with more than a few dozen parameters and rely on global surrogate models that scale cubically with the number of observations. In this work we propose MORBO, a scalable method for multi-objective BO over high-dimensional search spaces. MORBO identifies diverse globally optimal solutions by performing BO in multiple local regions of the design space in parallel using a coordinated strategy. We show that MORBO significantly advances the state-of-the-art in sample efficiency for several high-dimensional synthetic problems and real world applications, including an optical display design problem and a vehicle design problem with 146 and 222 parameters, respectively. On these problems, where existing BO algorithms fail to scale and perform well, MORBO provides practitioners with order-of-magnitude improvements in sample efficiency over the current approach.

4.2 Introduction

The challenge of identifying optimal trade-offs between multiple complex objective functions is pervasive in many fields, including machine learning [Sener and Koltun, 2018], science [Gopakumar et al., 2018], and engineering [Marler and Arora, 2004, Mathern et al., 2021]. For instance, Mazda recently proposed a vehicle design problem in which the goal is to optimize the widths of 222 structural parts in order to minimize the total weight of three different vehicles while simultaneously maximizing the number of common gauge parts [Kohira et al., 2018]. Additionally, this problem has 54 black-box constraints that enforce important performance requirements such as collision safety. Evaluating a design requires either crash-testing a physical prototype or running computationally demanding simulations. In fact, the original problem was solved on what at the time was the world's fastest supercomputer and took around 3,000 CPU years to compute [Oyama et al., 2017]. Another example is designing optical components for AR/VR applications, which requires optimizing complex geometries described by hundreds of parameters in order to identify designs that yield optimal trade-offs between image quality and efficiency of the optical device. Evaluating a design involves either fabricating and measuring prototypes or running computationally intensive simulations. For such problems, sample-efficient optimization is paramount.

Bayesian optimization (BO) has emerged as an effective, general, and sample-efficient approach for "black-box" optimization [Jones et al., 1998] and is highly effective for machine learning hyperparameter tuning [Turner et al., 2021]. However, in its basic form, BO is subject to important limitations. In particular, (i) successful applications typically consider low-dimensional search spaces, usually with less than 20 tunable parameters [Frazier, 2018], (ii) inference with the typical Gaussian Process (GP) surrogate models incurs cubic time complexity with respect to the number of data points, which prevents usage in the large-sample regime that is often necessary for high-dimensional problems, and (iii) most methods focus on single objective unconstrained problems. As a result, BO cannot easily be applied to either of the aforementioned Mazda vehicle design or the AR/VR optical design

problems. Moreover, high dimensional multi-objective problems requiring sample-efficient optimization are prevalent in many real-world settings such as groundwater remediation [Akhtar and Shoemaker, 2015], cell network configuration [Dreifuerst et al., 2021], and water resource management [Bai et al., 2017]. The state-of-the-art approach for this class of problems is NSGA-II [Deb et al., 2002], a popular evolutionary strategy, but with poor sample-efficiency, which hinders the progress of the scientists running these experiments.

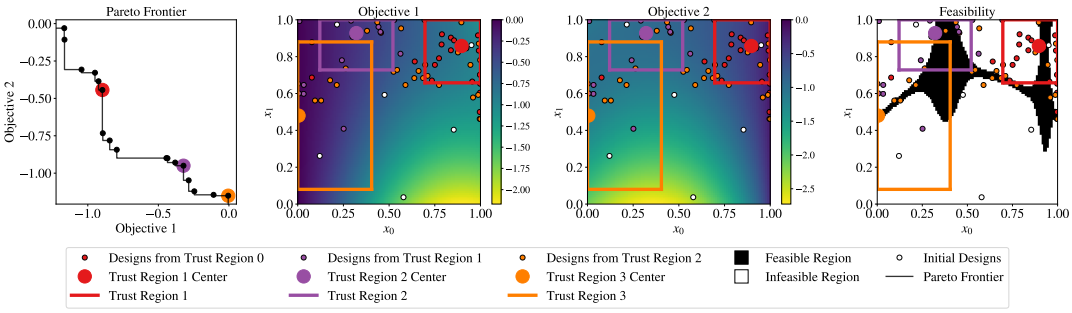

Figure 4.2.1: An illustration of MORBO on: 2-objective benchmark problem with 2 parameters and 2 constraints called MW7 [Ma and Wang, 2019] with 3 TRs. The left-most plot illustrates how MORBO's center selection technique centers the TRs at Pareto optimal points across different parts of the Pareto frontier. This encourages MORBO to explore diverse parts of the Pareto Frontier, which is important to identifying the multiple disconnected regions on this MW7 problem. The three right-most plots illustrate the TRs over the design space along with contours of, respectively, the 2 objective metrics and the feasibility metric indicating whether all black-box constraints are satisfied. Note that the TRs overlap with one another and contain data points that were collected by other TRs. Hence, sharing observations collected by different TRs provides local models with more observations than if each local model were only fitted to data collected by its corresponding TR.

In this paper, we close this gap by making BO applicable to challenging high-dimensional multi-objective problems. To do so, we propose an algorithm called MORBO ("Multi-Objective Regionalized Bayesian Optimization") that optimizes diverse parts of the global Pareto frontier in parallel using a coordinated set of local trust regions (TRs). As shown in Figure 4.2.1 (left), TRs are located at different solutions with diverse trade-offs between objectives. MORBO performs local BO in each TR to mitigate over-exploration, a phenomenon that plagues many algorithms in high-dimensional settings [Eriksson and Poloczek, 2021]. To enable scaling to large evaluation budgets, MORBO leverages *local* GP surrogate

models of the objective function, which reduces the time complexity for GP inference from $O(n^3)$, where n is the number of data points, to $O(n_{\mathcal{T}}^3)$, where $n_{\mathcal{T}} \ll n$ is the number of local data points for a TR $\mathcal{T}$. To facilitate efficient and collaborative global optimization, MORBO *passes information* between TRs in the following two ways: (1) Observations collected by one TR are shared with the others—which is particularly important when the TRs overlap as shown in Figure 4.2.1, (2) MORBO selects a batch of candidates by leveraging the TRs to collaboratively maximize a global utility. To ensure efficient global optimization, MORBO terminates under-performing TRs and allocates new TRs according to a global policy with a theoretical performance guarantee—a property that sets MORBO apart from most existing methods.

The significance of MORBO is that it is the *first* multi-objective BO method that scales to hundreds of tunable parameters and thousands of evaluations, a setting where practitioners have previously had to fall back on alternative methods with much lower sample-efficiency, such as NSGA-II. Our comprehensive evaluation demonstrates that MORBO yields *order-of-magnitude* savings in terms of time and resources compared to state-of-the-art methods on challenging high-dimensional multi-objective problems.

4.3 Background

4.3.1 Preliminaries

Multi-Objective Optimization

In multi-objective optimization (MOO), the goal is to maximize (without loss of generality) a vector-valued objective function $\boldsymbol{f}(\boldsymbol{x}) = [f^{(1)}(\boldsymbol{x}), ..., f^{(M)}(\boldsymbol{x})] \in \mathbb{R}^M$, where $M \geq 2$ while satisfying black-box constraints $\boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{0} \in \mathbb{R}^V$ where $V \geq 0$, $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$, and $\mathcal{X}$ is a compact set. Usually, there is no single solution $\boldsymbol{x}^*$ that simultaneously maximizes all M objectives and satisfies all V constraints. Hence, objective vectors are compared using Pareto domination.

Definition 4.3.1. An objective vector $\boldsymbol{f}(\boldsymbol{x})$ *Pareto-dominates* $\boldsymbol{f}(\boldsymbol{x}')$, denoted as $\boldsymbol{f}(\boldsymbol{x}) \succ \boldsymbol{f}(\boldsymbol{x}')$, if $f^{(m)}(\boldsymbol{x}) \geq f^{(m)}(\boldsymbol{x}')$ for all $m = 1, ..., M$ and there exists at least one $m \in \{1, \ldots, M\}$ such that $f^{(m)}(\boldsymbol{x}) > f^{(m)}(\boldsymbol{x}')$.

Definition 4.3.2. The *Pareto frontier* (PF) is the set of optimal trade-offs $\mathcal{P}(X)$ over a set of designs $X \subseteq \mathcal{X}$:

$$\mathcal{P}(X) = \{\boldsymbol{f}(\boldsymbol{x}) : \boldsymbol{x} \in X, \nexists\, \boldsymbol{x}' \in X \ s.t.\ \boldsymbol{f}(\boldsymbol{x}') \succ \boldsymbol{f}(\boldsymbol{x})\}$$

Under black-box constraints, the *feasible Pareto frontier* is defined as $\mathcal{P}_{\text{feas}}(X) = \mathcal{P}(\{\boldsymbol{x} \in X : \boldsymbol{g}(\boldsymbol{x}) \geq \boldsymbol{0}\})$.

The goal of a MOO algorithm is to identify an approximate PF $\mathcal{P}(X_n)$ of the true PF $\mathcal{P}(\mathcal{X})$ within a pre-specified budget of $|X_n| = n$ function evaluations. The quality of a PF is often evaluated using the hypervolume (HV) indicator.

Definition 4.3.3. The *hypervolume indicator*, $\text{HV}(\mathcal{P}(X)|\boldsymbol{r})$ is the M-dimensional Lebesgue measure λ_M of the region dominated by $\mathcal{P}(X)$ and bounded from below by a reference point $\boldsymbol{r} \in \mathbb{R}^M$.

The reference point is typically provided by the practitioner based on domain knowledge [Yang et al., 2019]. MOO problems are often addressed using evolutionary algorithms (EA) such as NSGA-II [Deb et al., 2002]. However, EAs generally suffer from high sample-complexity, rendering them inapplicable under small evaluation budgets.

Bayesian Optimization

When high sample-efficiency is required, Bayesian optimization (BO) is a popular approach [Frazier, 2018]. BO relies on a probabilistic surrogate model and an acquisition function that uses the surrogate model to provide the utility of evaluating a set of design points on the black-box function. The acquisition function is responsible for balancing exploration and exploitation. In the multi-objective setting, a common approach is to optimize random scalarizations of the objectives [Knowles, 2006, Paria et al., 2020] using a single-objective acquisition function. A more principled approach

is to directly optimize the Pareto frontier by selecting candidates with maximum hypervolume improvement either in expectation under the GP posterior [Emmerich et al., 2006] or using Thompson sampling (TS) [Bradford et al., 2018].

4.3.2 Related Work

Multi-objective Bayesian optimization

There have been many recent contributions to multi-objective BO, e.g., Konakovic Lukovic et al. [2020], Daulton et al. [2020, 2022a], Bradford et al. [2018]), but very few methods consider the high-dimensional setting and with large evaluation budgets. All of these methods described below rely on global GP models. As a result, these methods have mostly been evaluated on low-dimensional problems, typically $d \ll 10$ [Konakovic Lukovic et al., 2020, Bradford et al., 2018]. In the multi-objective BO literature, the largest search space we have found consists of 27 parameters [Paria et al., 2020]. Nevertheless, for completeness we review multi-objective BO methods that support generating large batches of designs. DGEMO [Konakovic Lukovic et al., 2020] uses a hypervolume-based objective with heuristics to encourage diversity while exploring the PF.

Parallel expected hypervolume improvement (qEHVI) [Daulton et al., 2020] has strong empirical performance, but its computational complexity scales exponentially with the batch size. qNEHVI [Daulton et al., 2021] improves scalability with respect to the batch size, but like DGEMO and qEHVI, qNEHVI has only been evaluated on low-dimensional search spaces. TSEMO [Bradford et al., 2018] optimizes approximate GP function samples using NSGA-II and uses a hypervolume-based objective for selecting a batch of points from the NSGA-II population. ParEGO [Knowles, 2006] and TS-TCH [Paria et al., 2020] use random Chebyshev scalarizations with parallel expected improvement [Jones et al., 1998] and Thompson sampling—where a design is sampled with probability proportional to a design being optimal [Thompson, 1933]—respectively. ParEGO has been extended to the batch setting in various ways including: (i) MOEA/D-EGO [Zhang et al., 2010], an algorithm that optimizes multiple scalarizations in parallel using MOEA/D [Zhou

et al., 2012], and (ii) qParEGO [Daulton et al., 2020], which uses composite objectives with sequential greedy batch selection under different scalarization weights. Information-theoretic methods, e.g., Hernandez-Lobato et al. [2016], Suzuki et al. [2020] have also garnered recent interest.

LaMOO [Zhao et al., 2022] is a recent work that partitions the search space into "good" and "bad" regions and samples new designs from "good" regions using qEHVI or CMA-ES [Hansen, 2007]. However, LaMOO-qEHVI relies on global GPs and is therefore prohibitively time-consuming with large evaluation budgets. In addition, the authors propose to use rejection sampling to enforce that samples are from the, typically non-rectangular, "good" region, but rejection sampling is prohibitively time-consuming in high-dimensional search spaces (see Appendix 4.D.1 for further discussion).

High-dimensional Bayesian optimization

Two popular approaches for high-dimensional BO are (1) mapping the high-dimensional inputs to a low-dimensional space via a random embedding [Wang et al., 2016b, Munteanu et al., 2019, Letham et al., 2020] and (2) exploiting additive structure [Kandasamy et al., 2015a, Gardner et al., 2017]. However, both families of methods require strong assumptions on the structure of the problem (low-dimensional linear or additive structure, respectively), and often perform poorly if the assumptions do not hold [Eriksson and Jankowiak, 2021]. This is especially problematic when optimizing multiple objectives since all objectives need to have the same assumed structure, which is unlikely in practice. Eriksson and Jankowiak [2021] leverage a weaker assumption that the objective only depends on a small subset of the parameters and Eriksson et al. [2021] extended this approach to the multi-objective setting, but this approach requires using computationally-demanding Markov Chain Monte Carlo methods for fitting the model, which is only feasible in the small data regime.

Trust Region Bayesian Optimization

Another popular method for high-dimensional BO is TuRBO [Eriksson et al., 2019], which performs BO in local trust regions (TRs) to avoid over-exploration. In contrast with [Zhao et al., 2022] which uses non-rectangular "good" regions, TuRBO uses hyperrectangular TRs, where each TR $\mathcal{T}$ has a center point $\boldsymbol{x}_{\text{center}}$ and an edge-length $L \in [L_{\min}, L_{\max}]$. Each TR maintains success and failure counters that record the number of consecutive samples generated from the TR that improved or failed to improve (respectively) the objective. If the success counter exceeds a predetermined threshold τ_{succ}, the TR length is increased to $\min\{2L, L_{\max}\}$ and the counter is reset to zero. Similarly, after τ_{fail} consecutive failures the TR length is set to $L/2$ and the failure counter is set to zero. Finally, if the length L drops below a minimum edge length $L_{\min}$, the TR is terminated and a new TR is initialized.

In contrast with aforementioned methods, TuRBO makes no strong assumptions about the objectives. Although TuRBO has been extended to handle black-box constraints [Eriksson and Poloczek, 2021], to our knowledge, all existing TR-based BO methods target single-objective optimization. In addition, TuRBO does not pass information between TRs, which results in an inefficient use of the evaluation budget; these methods have not observed significant improvement from using multiple TRs. Lastly, even though optimization is restricted to a local TR, TuRBO fits GP models to the entire history of data collected by a single TR which can lead to poor scalability in settings where TRs restart infrequently.

4.3.3 Issues with Scalarized TuRBO

Since ParEGO is a well-established method (in low-dimensional settings) that optimizes random Chebyshev scalarizations, a reasonable approach would be to extend TuRBO to the MOO setting by using multiple TRs in parallel where each TR optimizes a different random Chebyshev scalarization of the objectives. However, as we demonstrate in the left subplot of Figure 4.3.1, this approach results in a PF with very poor coverage. This is because a single scalarization is used for the lifetime of each TR in order to maintain a stable objective. Optimizing a single

scalarization per trust region often leads to better solutions with respect to that scalarization than optimizing the entire PF using a hypervolume-based acquisition functions, which requires exploration of different objective trade-offs. However, if TRs are not restarted frequently (e.g. because TuRBO continues to find better solutions with respect to that scalarization), only a small number of scalarizations will be used, which can lead to poor coverage of the PF. As shown in Figure 2, we observe that MORBO yields PFs with better coverage (diversity of trade-offs). In addition, the TRs in TuRBO are independent; they do not pass information about evaluated designs and observations, and they do not collaboratively aim to optimize the global PF—rather, they act in isolation to optimize their own objectives. Together, this leads to an inefficient use of the sample budget.

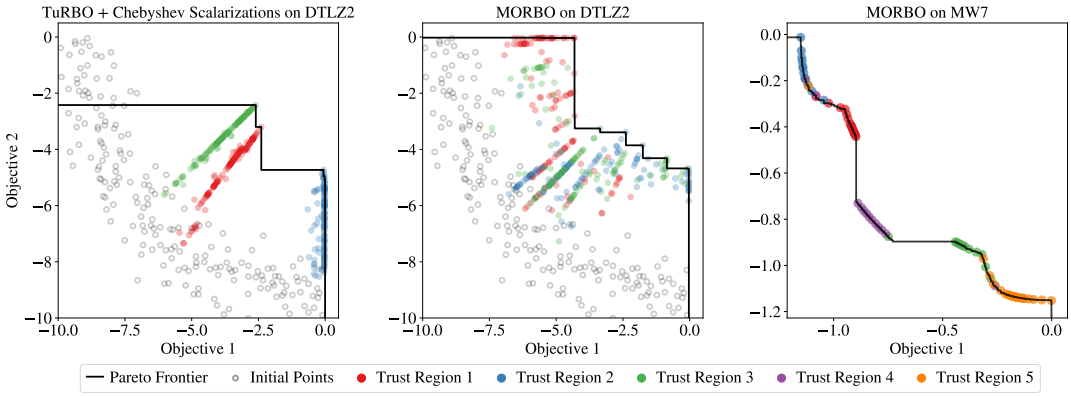

Figure 4.3.1: Objective values achieved on a 2-objective DTLZ2 function with $d = 100$ after 600 evaluations, batch size 50, and 3 TRs. The scatter plot illustrates the search behavior. The grey circles indicate the initial space-filling design, which is the same for both methods. The other marker shapes and colors indicate which of the 3 TRs obtained a given solution. The black line indicates the approximate Pareto frontier identified by each method. (Left) A straightforward extension of TuRBO where each TR optimizes a random Chebyshev scalarization of the objectives does not explore the trade-offs between the objectives because the TRs are rarely terminated under this approach, which leads only a few scalarizations being used. (Center) In contrast, MORBO employs a center selection strategy that actively targets under-explored regions of the Pareto frontier and uses a hypervolume-based acquisition function that is known to reward to high quality Pareto frontiers [Zitzler et al., 2003, Couckuyt et al., 2014, Yang et al., 2019] and explores the entirety of the PF. (Right) MORBO can discover disconnected regions of global PF on the MW7 function ($d = 10$, with 2 constraints) by using 5 TRs to locally optimize disjoint regions of PF collaboratively, in parallel. This is stark contrast with TuRBO with Chebyshev scalarizations which the left plot shows yield approximate Pareto frontiers with poor coverage and diversity, even when the true PF is connected and simple.

4.4 MORBO

We now introduce MORBO, a *collaborative* multi-TR approach for constrained high-dimensional multi-objective BO. Rather than following TuRBO's approach of employing multiple independent TRs, MORBO shares observations across TRs to provide each TR with all available information about the objectives and constraints relevant for local optimization in the TR. Moreover, MORBO further departs from TuRBO by (1) selecting TR center points in a coordinated fashion to encourage identifying Pareto frontiers with good coverage, (2) choosing new candidate designs by collaboratively optimizing a shared global utility, and (3) employing local models to reduce computational complexity and improve scalability in large data regimes. As shown in the center plot of Figure 4.3.1, MORBO identifies a high quality PF with much better coverage than the aforementioned simple TuRBO extension. For the remainder of this section, we describe the core components of MORBO, which are also summarized in Algorithm 1.

4.4.1 Collaborative Batch Selection via Global Utility Maximization

Maximizing hypervolume improvement (HVI) has been shown to produce high-quality and diverse PFs [Emmerich et al., 2006]. Given a reference point, the hypervolume improvement from a set of points is the increase in HV when adding these points to the previously selected points. Expected HVI (EHVI) is a popular acquisition function that integrates HVI over the GP posterior. However, maximizing EHVI directly requires re-computing the GP posterior and sampling from it in each gradient step, which becomes prohibitively slow as the number of objectives (and constraints) and in-sample data points increases.

To allow scalability to large batch sizes q, we instead use Thompson sampling (TS) to draw q posterior samples from the GP and optimize HVI under each realization. This approach can be viewed as a single-sample approximation of EHVI [Daulton et al., 2021]. We select q points $\boldsymbol{x}_1, ..., \boldsymbol{x}_q$ for the next batch in a *sequential greedy* fashion and condition upon the previously selected points in the batch by

Algorithm 1 Summary of MORBO

input Objective functions f, Number of trust region n_{TR}, Initial trust region length L_{init}, Maximum trust region length $L_{\max}$, Minimum trust region length $L_{\min}$.
output Approximate Pareto frontier $\mathcal{P}_n$

Evaluate an initial set of points and initialize the trust regions $\mathcal{T}_1, ..., \mathcal{T}_{n_{\text{TR}}}$ using the center selection procedure described in Section 4.4.2 and mark center points as unavailable for other trust regions.
$X_0 \leftarrow \emptyset, Y_0 = \emptyset, t \leftarrow 1$
while budget not exhausted **do**
  Fit a local model within each trust region.
  Select q candidates using the sequential greedy HVI procedure described in Section 4.4.1.
  Evaluate candidates on the true objective functions and obtain new observations.
  for $j = 1, ..., n_{\text{TR}}$ **do**
    Update trust regions with new observations as described in Section 4.4.
    Increment success/failure counters as described in Section 4.4 for observations from T_j.
    Update edgelength L_j for $\mathcal{T}_j$.
    if $L_j < L_{\min}$ **then**
      Terminate $\mathcal{T}_j$.
      Fit GP to restart points $\mathcal{D}_{t-1} = (X_{t-1}, Y_{t-1})$: $\boldsymbol{f}_{t-1} \sim P(\boldsymbol{f}|\mathcal{D}_{t-1})$.
      Sample $\boldsymbol{\lambda} \sim S_+^{M-1}$ and $\tilde{\boldsymbol{f}}_{t-1} \sim P(\boldsymbol{f}|\mathcal{D}_{t-1})$, where $S_+^{M-1} = \{\boldsymbol{w} \in \mathbb{R}_+^M : ||\boldsymbol{w}||_2 = 1\}$.
      $\boldsymbol{x}_t \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{X}} s_{\boldsymbol{\lambda}}[\tilde{\boldsymbol{f}}_{t-1}(\boldsymbol{x})]$, where $s_{\boldsymbol{\lambda}}[\boldsymbol{y}] = \min_m(\max(\frac{y_m}{\lambda_m}, 0))^M$ and $\cdot_i$ denotes the i^{th} element.
      Evaluate $\boldsymbol{x}_t$ on the objective functions and obtain new observation $\boldsymbol{y}_t$.
      Reinitialize $\mathcal{T}_j$ with edgelength L_{init} centered at the $\boldsymbol{x}_t$.
      Set $X_t \leftarrow X_{t-1} \cup \{\boldsymbol{x}_t\}, Y_t \leftarrow Y_{t-1} \cup \{\boldsymbol{y}_t\}$, $t \leftarrow t + 1$.
    end if
    Update center to the available point with maximum HVC (globally if $\mathcal{T}_j$ was terminated otherwise within $\mathcal{T}_j$).
  end for
end while

computing the HVI with respect to the current PF $\mathcal{P}$. In particular, to select the i^{th} point from a set of r candidate points $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_r$ we draw a sample from the joint posterior over $\boldsymbol{f}(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}\} \cup \{\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_r\})$, which yields the realization $\{\tilde{\boldsymbol{f}}(\boldsymbol{x}_1), \ldots, \tilde{\boldsymbol{f}}(\boldsymbol{x}_{i-1}), \tilde{\boldsymbol{f}}(\hat{\boldsymbol{x}}_1), \ldots, \tilde{\boldsymbol{f}}(\hat{\boldsymbol{x}}_r)\}$. We select the i^{th} point as the candidate point that maximizes the HVI jointly with the realizations $\tilde{\boldsymbol{f}}(\boldsymbol{x}_1), \ldots, \tilde{\boldsymbol{f}}(\boldsymbol{x}_{i-1})$ of the previously selected points as shown in Figure 4.A.1. Conditioning on the previously

selected points and computing the HVI under a sample from the joint posterior over the previously selected points and the discrete set of candidates leads to more diverse batch selection compared to selecting each point independently. Moreover, this approach effectively lets TRs collaboratively maximize the global HVI utility function. Using this global utility, an individual TR considers the iteration a success if at least one proposed candidate improves the global HV and a failure otherwise.

Another benefit of HV-based acquisition functions is that they naturally provide utility values for set of points, which enables the TRs to target different parts of the PF. This is particularly appealing in settings where the PF may be disjoint or may require exploring different parts of the search space. As shown in the right plot of Figure 4.3.1, MORBO recovers diverse regions of a disconnected PF. Lastly, we note that this batch selection strategy also allows to straightforwardly implement *fully asynchronous* optimization, where evaluations are dispatched to different "workers" and new candidates are generated whenever there is capacity in the worker pool. In the asynchronous setting, success/failure counters and TRs can be updated after every q observations are received, and intermediate observations can immediately be used to update the local models.

4.4.2 Coordinated Trust Region Center Selection

In (constrained) single-objective optimization, previous work centers the local TR at the best (feasible) observed point. However, in the multi-objective setting, there is typically no single best solution. Assuming noise-free observations, MORBO selects the center to be the feasible point on the PF with maximum hypervolume contribution (HVC) [Beume et al., 2007, Loshchilov et al., 2011]. If there is no feasible point, MORBO chooses the point with the smallest total constraint violation (see Appendix 4.B for details on center selection with constraints). Given a reference point, the HVC of a point on the PF is the reduction in HV if that point were to be removed; that is, the HVC of a point is its exclusive contribution to the PF. Centering a TR at the point with maximal HVC collected by that TR promotes coverage across the PF, as points in crowded regions will have lower contribution.

MORBO selects TR centers based on their HVCs in a sequential greedy fashion, excluding points that have already been selected as the center for another TR.

4.4.3 Local Modeling

Most BO methods use a single global GP model, often with a stationary kernel (e.g. Matérn-5/2) using automatic relevance determination (ARD) fitted to all observations collected so far. While a global model is necessary for most BO methods, MORBO only requires each model to be accurate within the corresponding TR. To increase scalability, we employ local modeling where we only include the observations contained within a local modeling hypercube with edge length $2L$. The motivation for using the observations from a slightly larger hypercube is to improve the model close to the TR boundary.

In previous trust region BO works [Eriksson et al., 2019, Eriksson and Poloczek, 2021, Wan et al., 2021], each TR uses a GP that is fitted to the all observations collected by that TR (rather than only a set of local observations in or near the TR), which leads to scalability issues due to the cubic time complexity of GP inference if the TR collects many observations. In addition, fitting a GP solely to data collected by a single TR ignores observations collected by other TRs and makes inefficient use of the sampling budget. In contrast, MORBO shares observations across TRs and employs local models, where models are fit to all observations within a hypercube with edge length $2L$. This significantly reduces the computational cost since exact GP fitting scales cubically with the number of data points. Under limited assumptions on the distribution of data across TRs, using local models results in speedups of $O(n_{\text{TR}}^2/\eta^3)$, where η is the average number of TR modeling spaces a data point resides in. Empirically, we demonstrate (see Figure 4.E.1 in Appendix 4.F) that $\eta < 1$ as the optimization progresses and the TRs shrink, and we find that this translates into speedups of two orders of magnitude relative to global modeling as shown in Appendix 4.F.2. See Appendix 4.E for more details on the complexity.

4.4.4 Re-initialization Strategy

Although MORBO performs local optimization within a TR, we ensure global optimization by re-initializing TRs using a principled technique based on hypervolume scalarizations [Zhang and Golovin, 2020]. A HV scalarization is defined as $s_{\boldsymbol{\lambda}}[\boldsymbol{y}] = \min_m(\max(\frac{y_m}{\lambda_m}, 0))^M$, where $\cdot_m$ denotes the m^{th} component [Zhang and Golovin, 2020]. Let $\mathcal{D}_{t-1} = (X_{t-1}, Y_{t-1})$ be the set of previous re-initialization (restart) points $X_{t-1} = \{\boldsymbol{x}_i\}_{i=1}^{t-1}$ and corresponding observations $Y_{t-1} = \boldsymbol{f}(X_{t-1})$, where $X_0 = \emptyset$ and $Y_0 = \emptyset$. Given $\mathcal{D}_{t-1}$, we determine the center point x_t of the new TR by maximizing a random HV scalarization of the objectives under a posterior sample from a global GP posterior conditioned on D_{t-1}: $\tilde{\boldsymbol{f}} \sim P(\boldsymbol{f}|\mathcal{D}_{t-1})$. This ensures that TRs are initialized in diverse parts of the objective space and yields a global optimization performance guarantee (Section 4.5).

4.5 Theoretical Analysis

We analyze the performance of MORBO in terms of its cumulative HV regret. The instantaneous HV regret $R(\mathcal{P}_t)$ after t TR restarts is defined as the difference in HV dominated by the true Pareto frontier $\mathcal{P}^*$ and the approximate Pareto frontier $\mathcal{P}_t$: $R(\mathcal{P}_t) = \text{HV}(\mathcal{P}^*) - \text{HV}(\mathcal{P}_t)$. The (cumulative) HV regret after T restarts is the sum of the instantaneous regret over all restarts: $R_T = \sum_{t=1}^{T} R(\mathcal{P}_t)$. First, we show that a TR will only evaluate a finite number of samples before restarting.

Lemma 4.5.1. *Let $\boldsymbol{f} \in [0, B]^M$, and assume that MORBO only considers a newly evaluated sample to be an improvement (for updating the corresponding TR's success and failure counters) if it increases the HV by at least $\delta \in \mathbb{R}^+$ and assume that success counter threshold $\tau_{succ} = \infty$.*[1] *Then each TR will only evaluate a finite number of samples.*

The proof is given in Appendix 4.C. Having established that TRs only evaluate a finite number of designs, we now bound the hypervolume regret with respect to the number of restarted TRs. The bound leverages the kernel-dependent

[1] As stated in Appendix 4.D, we use $\tau_{\text{succ}} = \infty$ in all of our experiments.

maximum information gain γ_T—which measures the decrease in uncertainty after T observations —and is commonly used to analyze regret in BO [Srinivas et al., 2010].

Theorem 4.5.1. *Let $\boldsymbol{f} \in [0, B]^M$ for $B > 0$ and let each component $f^{(m)}$ for $m = 1, ..., M$ follow a Gaussian distribution with marginal variances $\sigma \leq 1$ and independent observation noise $\epsilon_m \sim \mathcal{N}(0, \sigma_m^2)$ such that $\sigma_m^2 \leq \sigma^2 \leq 1$. Let $\mathcal{P}_t$ denote the Pareto frontier over $\boldsymbol{f}(X_t)$, where X_t is the set of TR re-initialization points after t TRs have been restarted. Suppose further that the conditions of Lemma 4.5.1 hold. Then, the cumulative hypervolume regret R_T of MORBO after T restarts is bounded by:*

$$R_T \leq M^2(\sqrt{2e\pi}B/2)^M\sqrt{d\gamma_T T \ln(T)}.$$

Up to logarithmic terms, this regret bound is on the order of $\tilde{\mathcal{O}}(\sqrt{T})$. This bound is significant because, to our knowledge, Zhang and Golovin [2020] is the only other work to bound the HV regret of multi-objective BO algorithms. This makes MORBO the first sample-efficient large-scale, MOO algorithm with bounded regret. The proof, given in Appendix 4.C, leverages the hypervolume regret bound from Zhang and Golovin [2020]. However, our regret bound is with respect to the number of restart points (rather than evaluations)—a difference that can be viewed as a cost of focusing on large-scale problems which BO with global GPs cannot address. Moreover, our regret analysis in terms of the number of restarts is similar to the convergence guarantees of gradient-based TR optimization methods [Yuan, 1999] and can be viewed as a multi-objective analogue of the performance guarantees of recent single-objective BO-based TR methods [Wan et al., 2021].

4.6 Experiments

We evaluate MORBO on an extensive suite of benchmarks with various numbers of input parameters (d), objectives (M), and constraints (V). In Appendix 4.F.1, we consider a vehicle ($d = 5$) and a welded beam ($d = 4$, $V = 4$) design problem to show that MORBO is competitive with other algorithms on problems it was not designed for. We consider three challenging real-world problems: a trajectory

planning problem ($d = 60$), a problem of designing optical systems for AR/VR applications ($d = 146$), and an automotive design problem ($d = 222, V = 54$) . In addition, we evaluate MORBO on DTLZ3, DTLZ5, and DTLZ7 problems with 2/4 objectives (6 problems in total) in Appendix 4.F.

We compare MORBO to multi-objective BO methods (qNEHVI, qParEGO, TS-TCH, TSEMO, DGEMO, MOEA/D-EGO), recent work leveraging search space partitioning (LaMOO-CMAES, LaMOO-qNEHVI), a widely used evolutionary algorithm (NSGA-II), and Sobol—a quasi-random baseline where designs are sampled from a scrambled Sobol sequence [Owen, 2003] (see Appendix 4.D for more details on the methods). MORBO is implemented using BoTorch [Balandat et al., 2020] and the code will be made publicly available soon. We run all methods for 20 replications and initialize them using the same quasi-random initial points for each replication. We use the same hyperparameters for MORBO on all problems and conduct analyze the sensitivity of MORBO to its hyperparameters in Figure 4.6.2. See Appendix 4.D for details on the experiment setup. All experiments used a Tesla V100 SXM2 GPU (16GB RAM).

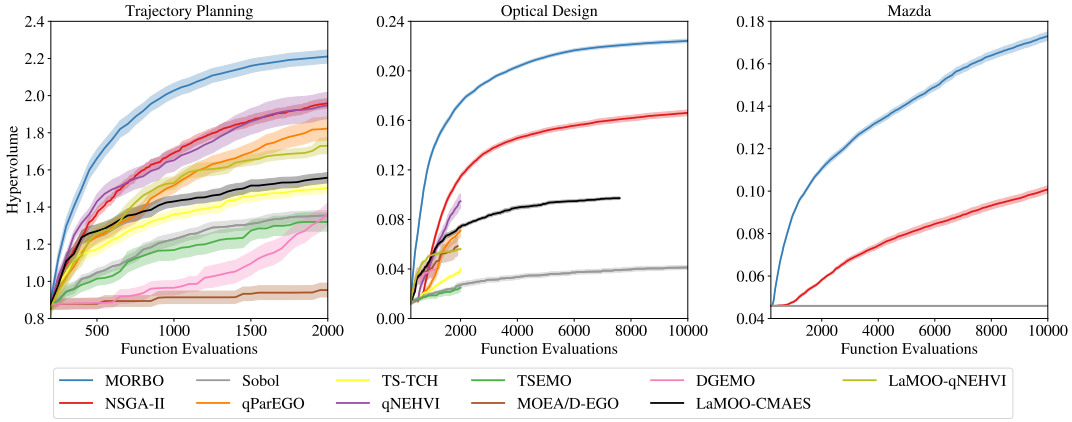

Figure 4.6.1: (Left) MORBO outperforms other methods on the trajectory planning problem ($d = 60$). (Middle) Illustration of the results on the Optical design problem ($d = 146$). NSGA-II performs better than the BO baselines but is not competitive with MORBO. (Right) MORBO shows compelling performance on the Mazda vehicle design problem ($d = 222$) with 54 black-box constraints. For all plots, we show the mean and one standard error of the mean over 20 replications.

4.6.1 Large-Scale Real-World Problems

Trajectory Planning We consider a trajectory planning problem similar to the rover trajectory planning problem considered in [Wang et al., 2018]. As in the original problem, the goal is to find a trajectory that maximizes the reward when integrated over the domain. The trajectory is determined by fitting a B-spline to 30 design points in the 2-objective plane, which yields a 60-dimensional optimization problem. In this experiment, we constrain the trajectory to begin at the pre-specified starting location, but we do not require it to end at the desired target location. In addition to maximize the reward of the trajectory, we also minimize the distance from the end of the trajectory to the intended target location. Intuitively, these two objectives are expected to be competing because reaching the exact end location may require passing through areas with lower associated reward. The results from 2,000 evaluations using batch size $q = 50$ and 200 initial points are presented in Figure 4.6.1, which shows that MORBO performs the best and even state-of-the-art methods such as qNEHVI do not out perform NSGA-II.

Optical design problem We consider the problem of designing an optical system for an augmented reality (AR) see-through display. This optimization task has 146 parameters describing the geometry and surface morphology of multiple optical elements in the display stack. Several objectives are of interest in this problem, including display efficiency and display quality. Each evaluation of these metrics requires a computationally intensive physics simulation that takes several hours to run. In this benchmark, the task is to explore the Pareto frontier between display efficiency and display quality (both objectives are normalized w.r.t. the reference point). We consider 250 initial points, batch size $q = 50$, and a total of 10,000 evaluations. This is out of reach for the other BO baselines due to runtime considerations, and so we run qNEHVI, qParEGO, TS-TCH, TSEMO, MOEA/D-EGO, for 2,000 evaluations and DGEMO for 1,000 evaluations. We were only able to run LaMOO-CMAES for $7,600$ evaluations before it overflowed GPU memory. Figure 4.6.1 shows that MORBO achieves substantial improvements

in sample efficiency compared to NSGA-II. Furthermore, observe that no other baselines are competitive with NSGA-II except in the very small sample regime (less than 500 evaluations).

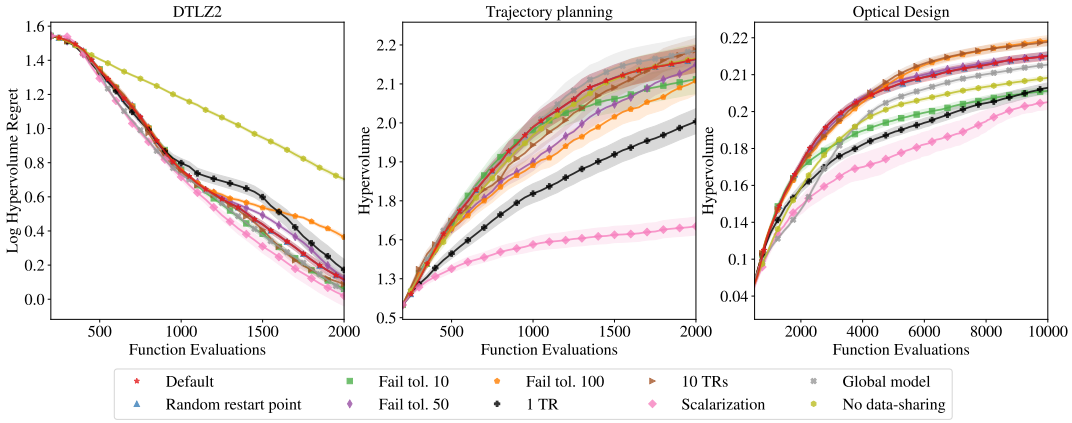

Figure 4.6.2: We investigate the sensitivity of MORBO with respect to its hyperparameters. We observe that using multiple TRs performs significantly better than using a single TR and that data-sharing and the use of a hypervolume based acquisition function are important components of MORBO.

Mazda vehicle design problem We consider the 3-car Mazda benchmark problem [Kohira et al., 2018]. This challenging MOO problem involves tuning 222 decision variables that represent the thickness of different structural parts. The goal is to minimize the total vehicle mass of the three vehicles (Mazda CX-5, Mazda 6, and Mazda 3) as well as maximizing the number of parts shared across vehicles. Additionally, there are 54 black-box output constraints (evaluated jointly with the two objectives) that enforce that designs meet performance requirements such as collision safety standards. This problem is, to the best our knowledge, the largest MOO problem considered by any BO method and requires fitting 56 GP models to the objectives and constraints. The original problem underlying the Mazda benchmark was solved on what at the time was the world's fastest supercomputer and took around 3,000 CPU years to compute [Oyama et al., 2017]. We consider a budget of 10,000 evaluations using batches of size $q = 50$ and 300 initial points.

Figure 4.6.1 demonstrates that MORBO clearly outperforms the other methods. A feasible design satisfying the black-box constraints was provided to all methods

for all replications as part of the initial 300 design points. However, in subsequent evaluations Sobol did not find another feasible design, illustrating the challenge of satisfying the 54 constraints. While NSGA-II made progress from the initial feasible solution, it is not competitive with MORBO. NSGA-II and Sobol are the only applicable baselines because standard multi-objective BO methods are impractically slow with 56 *global* GPs and LaMOO does not support black-box constraints.

4.6.2 Ablation study

Finally, we study the sensitivity of MORBO with respect to the number of TRs (n_{TR}), the failure tolerance (τ_{fail}), and sharing observations across TRs, local modeling, HVI acquisition function, and the re-initialization strategy. Using several TRs allows MORBO to explore different parts of the search space that potentially contribute to different parts of the Pareto frontier. The failure tolerance controls how quickly each TR shrinks: A large τ_{fail} leads to slow shrinkage and potentially too much exploration, while a small τ_{fail} may cause each TR to shrink too quickly and not collect enough data. MORBO uses 5 TRs and $\tau_{\text{fail}} = \max(10, \frac{d}{3})$ by default, similar to what is used by Eriksson et al. [2019].

We consider the DTLZ2 problem ($d = 100$, $M = 2$), the trajectory planning problem ($d = 60$, $M = 2$), and the optical design problem ($d = 146$, $M = 2$). Figure 4.6.2 shows that MORBO with the default settings performs well on all three problems. We observe that multiple TRs and the HVI acquisition function are important as neither a single TR nor a Chebyshev scalarization performs well. The performance of MORBO is robust to the choice of failure tolerance except for on the optical design problem where using a value of 10 is clearly worse than the default and causes the TRs to shrink too quickly. Not sharing data between TRs results in inferior results on the DTLZ2 and optical design problems. While using a global GP model achieves good results on the DTLZ2 and trajectory planning problems, it does not perform as well on the optical design problem. A global GP also comes at a high computational cost. Using a global GP, running MORBO with a budget of 10,000 evaluations on the optical design problem required 30 hours

of computational overhead, whereas MORBO did 10,000 evaluations in less than an hour using local models. Lastly, we find consistently strong performance for both our default HV scalarization-based re-initialization strategy and a strategy that selects a new design at random (denoted as "Random restart points"). The former allows us to bound MORBO's regret.

4.7 Discussion

We proposed MORBO, an algorithm for multi-objective BO over high-dimensional search spaces. By using a coordinated, collaborative multi-trust-region approach with scalable local modeling, MORBO scales gracefully to high-dimensional problems and high-throughput settings. In a comprehensive experimental evaluation, we showed that MORBO allows us to *effectively tackle important real-world problems that were previously out of reach for existing BO methods.* We showed that MORBO achieves substantial improvements in sample efficiency compared to existing state-of-the-art methods such as evolutionary algorithms. Due to the lack of alternatives, NSGA-II has been the method of choice for many practitioners, and we expect MORBO to provide practitioners with significant savings in terms of time and resources across the many disciplines that require solving challenging optimization problems.

However, there are some limitations to our method. Although MORBO can handle a large number of black-box constraints, using hypervolume-based acquisition means the computational complexity scales poorly with the number of objectives. Furthermore, MORBO is optimized for the large-batch high-throughput setting and other methods may be more suitable for and achieve better performance on low-dimensional problems with small evaluation budgets.

Appendix

4.A Details on Batch Selection

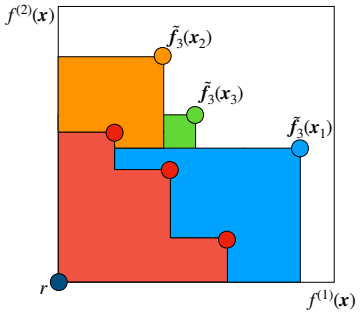

Figure 4.A.1: A visualization of our batch selection using HVI with $q = 4$. The red points represent the current PF. Blue, orange, and green points show the function values for the 3 selected points under the next posterior sample. To select the 4th point, the HVI of each candidate is evaluated jointly with the red, blue, orange, and green points.

As discussed in Section 4.3, over-exploration can be an issue in high-dimensional BO because there is typically high uncertainty on the boundary of the search space, which often results in over-exploration. This is particularly problematic when using continuous optimization routines to find the maximizer of the acquisition function since the global optimum of the acquisition function will often be on the boundary, see Oh et al. [2018] for a discussion on the "boundary issue" in BO. While the use of trust regions alleviates this issue, this boundary issue can still be problematic, especially when the trust regions are large.

To mitigate this issue of over-exploration, we use a discrete set of candidates by perturbing randomly sampled Pareto optimal points within a trust region by replacing only a small subset of the dimensions with quasi-random values from a scrambled Sobol sequence. This is similar to the approach used by Eriksson and Poloczek [2021] which proved crucial for good performance on high-dimensional

problems. In addition, we also decrease the perturbation probability p_n as the optimization progresses, which Regis and Shoemaker [2013] found to improve optimization performance. The perturbation probability p_n is set according to the following schedule:

$$p_n = p_0 \left[1 - 0.5 \frac{\log n'}{\log b} \right],$$

where n_0 is the number of initial points, n_f is the total evaluation budget, $p_0 = \min\{\frac{20}{d}, 1\}$, $b = n_f - n_0$, and $n' = \min\{\max\{n - n_0, 1\}, b\}$.

Given a discrete set of candidates, MORBO draws samples from the joint posterior over the function values for the candidates in this set and the previously selected candidates in the current batch, and selects the candidate with maximum HVI across the joint samples. This procedure is repeated to build the entire batch.[2] Using standard Cholesky-based approaches, exact posterior sampling has complexity that is cubic with respect to the number of test points and therefore is only feasible for relatively small discrete sets.

4.A.1 RFFs for fast posterior sampling

While asymptotically faster approximations than exact sampling exist; see Pleiss et al. [2020] for a comprehensive review, these methods still limit the candidate set to be of modest size (albeit larger), which may not do an adequate job of covering a the entire input space. Among the alternatives to exact posterior sampling, we consider using Random Fourier Features (RFFs) [Rahimi and Recht, 2008], which provide a deterministic approximation of a GP function sample as a linear combination of Fourier basis functions. This approach has empirically been shown to perform well with Thompson sampling for multi-objective optimization [Bradford et al., 2018]. The RFF samples are cheap to evaluate and which enables using much larger discrete sets of candidates since the joint posterior over the discrete set does not need to be computed. Furthermore, the RFF samples are differentiable with respect to the new candidate $\boldsymbol{x}$, and HVI is differentiable with respect to $\boldsymbol{x}$ using cached

[2] In the case that the candidate point does not satisfy that satisfy all outcome constraints under the sampled GP function, the acquisition value is set to be the negative constraint violation.

box decompositions [Daulton et al., 2021], so we can use second-order gradient optimization methods to maximize HVI under the RFF samples.

We tried to optimize these RFF samples using a gradient based optimizer, but found that many parameters ended up on the boundary, which led to over-exploration and poor BO performance. In an attempt to address this over-exploration issue, we instead consider continuous optimization over axis-aligned subspaces which is a continuous analogue of the discrete perturbation procedure described in the previous section. Specifically, we generate a discrete set of candidates points by perturbing random subsets of dimensions according to p_n, as in the exact sampling case. Then, we take the top 5 initial points with the maximum HVI under the RFF sample. For each of these best initial points we optimize only over the perturbed dimensions using a gradient based optimizer.

Figure 4.A.2 shows that the RFF approximation with continuous optimization over axis-aligned subspaces works well on for $D = 10$ on the DTLZ2 function, but the performance degrades as the dimensionality increases. Thus, the performance of MORBO can likely be improved on low-dimensional problems by using continuous optimization; we used exact sampling on a discrete set for all experiments in the paper for consistency. We also see that as the dimensionality increases, using RFFs over a discrete set achieves better performance than using continuous optimization. In high-dimensional search spaces, we find that exact posterior sampling over a discrete set achieves better performance than using RFFs, which we hypothesize is due to the quality of the RFF approximations degrading in higher dimensions. Indeed, as shown in Figure 4.A.2, optimization performance using RFFs improves if we use more basis functions on higher dimensional problems (4096 works better than 1024).

4.B Additional details of constraint handling in MORBO

If there are feasible points, the center is selected as the point with maximum HVC across the feasible Pareto frontier. If there are no feasible points, the center is

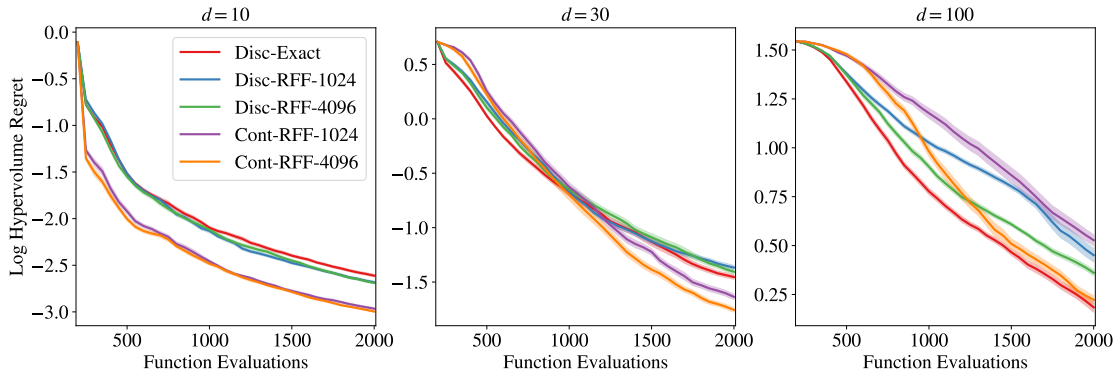

Figure 4.A.2: Optimization performance under various Thompson sampling approaches on DTLZ2 test problems with 2 objectives and various input dimensions $d \in \{10, 30, 100\}$. Disc-Exact uses exact samples from the joint posterior over a discrete set of 4096 points. Disc-RFF-1024 and Disc-RFF-4096 evaluate approximate sample paths (RFFs) over a discrete set of 4096 points with 1024 and 4096 basis functions, respectively. Cont-RFF-1024 and Cont-RFF-4096 use L-BFGS-B with exact gradients to optimize RFF draws along a subset of the dimensions (see in Appendix 4.A.1 for details) using 1024 and 4096 basis functions, respectively.

selected to be the point with minimum total constraint violation (the sum of the constraint violations). A TR's success counter is incremented if the TR center was feasible and the candidates generated from this TR improved the feasible hypervolume or if the TR center was infeasible and a candidate generated from this TR has lower total constraint violation than the TR center.

4.C Proofs

Lemma 4.1. Let $\boldsymbol{f} \in [0, B]^M$, and assume that MORBO only considers a newly evaluated sample to be an improvement (for updating the corresponding TR's success and failure counters) if it increases the HV by at least $\delta \in \mathbb{R}^+$ and assume that success counter threshold $\tau_{\text{succ}} = \infty$.[3] Then each TR will only evaluate a finite number of samples.

Proof. First, note that the hypervolume of the true Pareto frontier $\mathcal{P}^*$ is bounded. Without loss of generality, if the reference point $\boldsymbol{r} = \mathbf{0}$, then the $\text{HV}(\mathcal{P}^*) \leq B^M$. Suppose that a trust region evaluates an infinite number of samples. Then, the trust region has not had $1 + \log_2 L_{\text{init}} - \log_2 L_{\text{min}}$ streaks of τ_{fail} consecutive

[3] As stated in Appendix 4.D, we use $\tau_{\text{succ}} = \infty$ in all of our experiments.

failures. Hence, the trust region has increased the hypervolume of the Pareto frontier over the previously evaluated designs by at least δ infinitely many times. Hence, the hypervolume over the previously evaluated designs is infinite. This is a contradiction. □

Theorem 4.1. Let $\boldsymbol{f} \in [0, B]^M$ for $B > 0$ and let each component $f^{(m)}$ for $m = 1, ..., M$ follow a Gaussian distribution with marginal variances $\sigma \leq 1$ and independent observation noise $\epsilon_m \sim \mathcal{N}(0, \sigma_m^2)$ such that $\sigma_m^2 \leq \sigma^2 \leq 1$. Let $\mathcal{P}_t$ denote the Pareto frontier over $\boldsymbol{f}(X_t)$, where X_t is the set of TR re-initialization points after t TRs have been restarted. Suppose further that the conditions of Lemma 4.5.1 hold. Then, the cumulative hypervolume regret R_T of MORBO after T restarts is bounded by:

$$R_T \leq M^2(\sqrt{2e\pi}B/2)^M\sqrt{d\gamma_T T \ln(T)}.$$

Proof. From Lemma 4.5.1, we have that each trust region will only evaluate a finite number of samples. Hence, as the number of evaluations goes to infinity, MORBO will terminate and select new initial center points for trust regions an infinite number of times. Our regret bound is in terms of the number of restart points.

Our proof follows that of Zhang and Golovin [2020, Theorem 8], but the final form of our bound holds for arbitrary B. Note that lines 13-19 in Algorithm 1 correspond to Paria et al. [2020, Algorithm 1] using Thompson sampling, where the only evaluations are the $t-1$ restart points. From Paria et al. [2020, Theorem 1], the scalarized Bayes regret of Paria et al. [2020, Algorithm 1] using L–Lipschitz scalarizations is $O\Big(LMd^{\frac{1}{2}}[\gamma_T T \ln(T)]^{\frac{1}{2}}\Big)$. Since a hypervolume scalarization $s_{\boldsymbol{\lambda}}[\boldsymbol{y}]$ is $\mathcal{O}(B^M M^{1+M/2})$–Lipschitz [Zhang and Golovin, 2020, Lemma 6], we have that $L \leq B^M M^{1+M/2}$. From Zhang and Golovin [2020, Proof of Theorem 8], the hypervolume regret can be expressed by scaling the scalarized Bayes regret by a constant $c_M = \frac{\pi^{\frac{M}{2}}}{2^M \Gamma(\frac{M}{2}+1)}$ that depends on the number of objectives. Hence, we can

bound the hypervolume regret as:

$$R_T = \sum_{t=1}^{T} \mathrm{HV}(\mathcal{P}^*) - \mathrm{HV}(\mathcal{P}_t) \leq c_M L M d^{\frac{1}{2}} [\gamma_T T \ln(T)]^{\frac{1}{2}}.$$

Note that

$$c_M L \leq B^M M^{1+M/2} \frac{\pi^{\frac{M}{2}}}{2^M \Gamma(\frac{M}{2}+1)}$$

From Li and Chen [2007, Theorem 1], $\Gamma(x) > \frac{x^{x-\gamma}}{e^{x-1}}$, where $\gamma \approx 0.577$ is the Euler-Mascheroni constant. So,

$$\Gamma(M/2+1) > \frac{(M/2+1)^{(M/2+1-\gamma)}}{e^{(M/2)}} > \frac{M^{(M/2)}}{2e^{(M/2)}}.$$

Hence,

$$\frac{1}{\Gamma(\frac{M}{2}+1)} < \frac{(2e)^{(M/2)}}{M^{(M/2)}}.$$

So,

$$\begin{aligned} c_M L &\leq B^M M^{1+M/2} \frac{\pi^{\frac{M}{2}}}{2^M \Gamma(\frac{M}{2}+1)} \\ &\leq B^M M \frac{(2e\pi)^{\frac{M}{2}}}{2^M} \\ &\leq M\Big(\sqrt{2e\pi} B/2\Big)^M. \end{aligned}$$

So the cumulative regret bound is

$$\begin{aligned} R_T &\leq c_M L M d^{\frac{1}{2}} [\gamma_T T \ln(T)]^{\frac{1}{2}} \\ &\leq M^2 (\sqrt{2e\pi} B/2)^M d^{\frac{1}{2}} [\gamma_T T \ln(T)]^{\frac{1}{2}}. \end{aligned}$$

□

4.D Details on Experiments

4.D.1 Algorithmic details

For MORBO, we use 5 trust regions, which we observed was a robust choice in Figure 4.6.2. Following [Eriksson et al., 2019], we set $L_{\text{init}} = 0.8$, $L_{\max} = 1.6$, and use a minimum length of $L_{\min} = 0.01$. We use 4096 discrete points for optimizing

HVI for the vehicle safety and welded beam problems, 2048 discrete points on the trajectory planning and optical design problems, and 512 discrete points on the Mazda problem. Note that while the number of discrete points should ideally be chosen as large as possible, it offers a way to control the computational overhead of MORBO; we used a smaller value for the Mazda problem due to the fact that we need to sample from a total of 56 GP models in each trust region as there are 54 black-box constraints. We use an independent GP with a a constant mean function and a Matérn-5/2 kernel with automatic relevance detection (ARD) and fit the GP hyperparameters by maximizing the marginal log-likelihood (the same model is used for all BO baselines).

When fitting a model for MORBO, we include the data within a hypercube around the trust region center with edgelength $2L$. In the case that there are less than $N_m := \min\{250, 2d\}$ points within that region, we include the N_m closest points to the trust region center for model fitting. The success streak tolerance is set to be infinity, which prevents the trust region from expanding; we find this leads to good optimization performance when data is shared across trust regions. For qNEHVI and qParEGO, we use 128 quasi-MC samples and for TS-TCH, we optimize RFFs with 500 Fourier basis functions. All three methods are optimized using L-BFGS-B with 20 random restarts. For DGEMO, TSEMO, and MOEA/D-EGO, we use the default settings in the open-source implementation at `https://github.com/yunshengtian/DGEMO/tree/master`. Similarly, we use the default settings for NSGA-II the Platypus package (`https://github.com/Project-Platypus/Platypus`). We encode the reference point as a black-box constraint to provide this information to NSGA-II.

LaMOO in High-Dimensional Search Spaces

For LaMOO methods, leverage the implementation of LaMOO available at `https://drive.google.com/drive/folders/1CMdg5iBdbKe3nkboIjiS998rnBEV09EB?usp=sharing`. We set the exploration parameter C_p dynamically using the heuristic proposed by Zhao et al. [2022] to be 10% of the hypervolume of the current Pareto

frontier over the previously evaluated designs. We follow Zhao et al. [2022] and set the minimum leaf sample size to be 10.

Zhao et al. [2022] propose to use qEHVI with LaMOO, but we opt to use qNEHVI instead since it is capable of scaling to the batch size of $q = 50$ used in many of our experiments. We refer to this method as LaMOO-qNEHVI. We note that qNEHVI is mathematically equivalent to qEHVI on noiseless problems. The authors propose using rejection sampling to ensure samples come from the "good" region. For high-dimensional search spaces, the acceptance probability is low for uniform random samples from the global design space, and therefore, rejection sampling is prohibitively slow. Rejection sampling is used 1) to select starting points for multi-start L-BFGS-B and within the L-BFGS-B routine to enforce that samples are within the "good" region. We contacted the authors about computational issues with this approach, and the authors recommended to use rejection sampling for selecting starting points, and then to simply run L-BFGS-B from these "good" starting points across the global search space. With this approach, the resulting candidates may not (and often are not) within the "good" region, and LaMOO-qNEHVI is simply an initialization heuristic for optimizing qNEHVI, but this approach does speed up candidate generation quite a bit. Nevertheless, even using rejection sampling to generate starting points for L-BFGS-B can be (and is on our problems) prohibitively expensive in high-dimensional search spaces. Hence, we limit the rejection sampling by only considering $120,000$ design points before beginning L-BFGS-B with the most promising designs (whether or not they are in the "good" region). This makes LaMOO-qNEHVI feasible to run our our high-dimensional problems.

For LaMOO-CMA-ES, we use $q = 5$ rather than $q = 1$ on vehicle safety, as $q = 1$ is not supported.

4.D.2 Synthetic problems

The reference points for all problems are given in Table 4.D.1. We multiply the objectives (and reference points) for all synthetic problems by -1 and maximize the resulting objectives.

Problem	Reference Point
DTLZ2	[6, 6]
DTLZ3	$[10^3]^M$
DTLZ5	$[10]^M$
DTLZ7	$[15]^M$
Vehicle Safety	[1698.55, 11.21, 0.29]
Welded Beam	[40, 0.015]
MW7	[1.2, 1.2]

Table 4.D.1: The reference points for each synthetic benchmark problem.

DTLZ: We consider the 2-objective DTLZ2 problem with various input dimensions $d \in \{10, 30, 100\}$. We also use 2-objective and 4-objective variants of DTLZ3, DTLZ5, and DTLZ7 with $d = 100$. The DTLZ problems are standard test problems from the multi-objective optimization literature. Mathematical formulas for the objectives in each problem are given in Deb et al. [2002].

MW7: For a second test problem from the multi-objective optimization literature, we consider a MW7 problem with 2 objectives, 2 constraints, and $d = 10$ parameters. See Ma and Wang [2019] for details.

Welded Beam: The welded beam problem [Ray and Liew, 2002] is a structural design problem with $d = 4$ input parameters controlling the size of the beam where the goal is to minimize 2 objectives (cost and end deflection) subject to 4 constraints. More details are given in Tanabe and Ishibuchi [2020].

Vehicle Safety: The vehicle safety problem is a 3-objective problem with $d = 5$ parameters controlling the widths of different components of the vehicle's frame. The goal is to minimize mass (which is correlated with fuel economy), toe-box intrusion (vehicle damage), and acceleration in a full-frontal collision (passenger injury). See Tanabe and Ishibuchi [2020] for additional details.

4.D.3 Trajectory planning

For the trajectory planning, we consider a trajectory specified by 30 design points that starts at the pre-specified starting location. Given the 30 design points, we fit a B-spline with interpolation and integrate over this B-spline to compute the final reward using the same domain as in Wang et al. [2018]. Rather than directly optimizing the locations of the design points, we optimize the difference (step) between two consecutive design points, each one constrained to be in the domain $[0, 0.05] \times [0, 0.05]$. We use a reference point of [0, 0.5], which means that we want a reward larger than 0 and a distance that is no more than 0.5 from the target location [0.95, 0.95]. Since we maximize both objectives, we optimize the distance metric and the corresponding reference point value by -1.

4.D.4 Optical design

In order to obtain precise estimates of the optimization performance at reasonable computational cost, we conduct our evaluation on a neural network surrogate model of the optical system rather than on the actual physics simulator. The surrogate model was constructed from a dataset of 101,000 optical designs and resulting display images to provide an accurate representation of the real problem. The surrogate model is a neural network with a convolutional autoencoder architecture. The model was trained using 80,000 training examples and minimizing MSE (averaged over images, pixels, and RGB color channels) on a validation set of 20,000 examples. A total of 1,000 examples were held-out for final evaluation.

4.D.5 Mazda vehicle design problem

We follow the suggestions by Kohira et al. [2018] and use the reference point $[1.1, 0]$ and optimize the normalized objectives $\tilde{f}_1 = f_1 - 2$ and $\tilde{f}_2 = f_2/74$ corresponding to the total mass and number of common gauge parts, respectively. Additionally, an initial feasible point is provided with objective values $f_1 = 3.003$ and $f_2 = 35$, corresponding to an initial hypervolume of ≈ 0.046 for the normalized objectives. This initial solution is given to all algorithms. We limit the number of points used

for model fitting to only include the 2,000 points closest to the trust region center in case there are more than 2,000 in the larger hypercube with side length $2L$. Still, for each iteration MORBO using 5 trust regions fits a total of 56×5 GP models, a scale far out of reach for any other multi-objective BO method.

4.E Complexity Improvements from Local Modeling

The differences in model fitting time can be even more profound. To illustrate this, consider a situation in which a total of N data points have been collected by n_{TR} trust regions. Suppose for simplicity that each TR has the same number of observations (under some abuse of nomenclature we use TR to refer to the modeling domain of a TR in this section). Let η denote the average number of trust regions that a data point is part of. Then the number of points in each TR is $\eta N / n_{\text{TR}}$. Assuming cubic time complexity for model fitting (i.e. $O(N^3)$ if we used a single global model), the total time complexity of fitting all n_{TR} models in the individual TRs is $O\Big(n_{\text{TR}}(\eta N/n_{\text{TR}})^3\Big) = O\Big(\eta^3 N^3 / n_{\text{TR}}^2\Big)$. This will lead to asymptotic speedups of order $O\Big(n_{\text{TR}}^2/\eta^3\Big)$ when using local modeling. Typically, as the optimization progresses and the trust regions shrink, η becomes quite small (e.g. $\eta < 1$)[4]. We validate this claim empirically in the lower right subplot in Figure 4.E.1, which shows that η becomes less than 1 on the all problems considered as the optimization progresses. In Figure 4.E.1 we illustrate some additional information from the trust regions to better understand the role of data-sharing and local modeling in MORBO. Thus, the speedup relative to fitting a single global model can be multiple orders of magnitude.

4.E.1 Model fitting times

Empirically, we verify this speedup in Figure 4.E.2. This can also be seen in the

[4]When η is close to the number of trust regions, the "local" models will fit to nearly all observations, and hence, the models will essentially be global models. The value of η at the start of the optimization depends on the initial trust region edge length and the dimension of the search space.

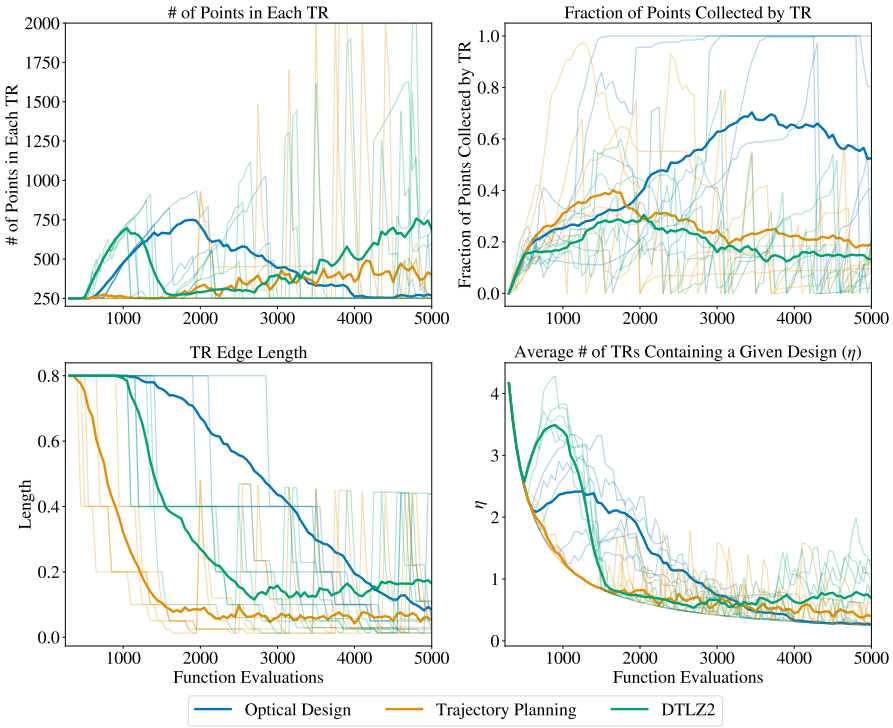

Figure 4.E.1: For the optical design, trajectory planning, and DTLZ2 problems. We show the average across replications as a solid line and traces from the first replication as transparent lines. (Upper Left) The number of points in each trust region. Trust regions often usually have a few hundred points on average, which results in computationally efficient local modeling. (Upper Right) The number of points in a trust region that was collected by that trust region. This shows that a large fraction of data within a trust region was actually collected by another trust region. (Lower Left) The trust region length. As the optimization proceeds, the trust regions shrink to focus on specific parts of the search space. (Lower Right) The average number of TRs that contain a given design, $\eta \in [0, N_{\text{TR}}]$. This shows that as the optimization progresses and the TRs shrink, on average less than 1 TR contains a given design. This is empirical validation of the claim in Appendix 4.E that η typically becomes small as the optimization progresses and therefore, the complexity improvements are substantial.

results in Tables 4.E.2 and 4.E.1. While candidate generation is fast for TSEMO, the model fitting causes a significant overhead with almost an hour being spent on model fitting after collecting 2,000 evaluations on the trajectory planning problem. This is significantly longer than for MORBO, which only requires far less time for the model fitting due to the use of local modeling. This shows that the use of local modeling is a crucial component of MORBO that limits the computational overhead from the model fitting. The model fitting for MORBO on the optical design problem is less than 25 seconds while methods such as DGEMO and TSEMO that rely on

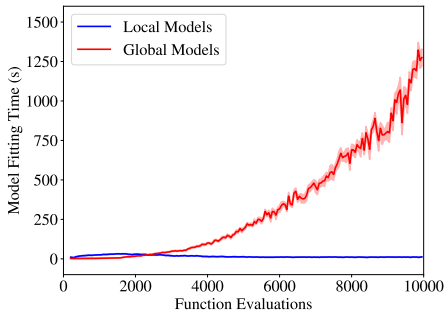

Figure 4.E.2: Model fitting time for MORBO with local modeling compared to MORBO with one global model on the 146-dimensional optical design problem. Fitting a global model takes almost 20 minutes towards the end of the optimization run compared to 10 seconds for MORBO.

global modeling require far more time for model fitting after only collecting 1,200 points. Additionally, while MORBO needs to fit as many as $56 \times 5 = 280$ GP models on the Mazda problem due to the 54 black-box constraints and the use of 5 trust regions, the total time for model fitting still is less than 3 minutes while this problem is completely out of reach for the other BO methods that rely on global modeling.

Problem	DTLZ3 ($M = 2$)	DTLZ5 ($M = 2$)	DTLZ7 ($M = 2$)	DTLZ3 ($M = 4$)	DTLZ5 ($M = 4$)	DTLZ7 ($M = 4$)
MORBO	11.0 (0.6)	9.7 (0.4)	10.6 (0.4)	11.5 (0.9)	10.5 (0.5)	10.6 (0.4)
NSGA-II	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
qParEGO	139.5 (24.6)	49.1 (2.2)	26.0 (2.5)	137.2 (15.4)	113.2 (6.6)	49.0 (3.5)
TS-TCH	64.5 (3.4)	93.9 (5.8)	89.6 (3.5)	143.3 (5.9)	167.6 (8.8)	141.3 (6.1)
qNEHVI	133.2 (23.9)	48.9 (4.9)	20.8 (1.7)	25.9 (2.3)	19.8 (1.7)	6.8 (0.4)
DGEMO	5425.1 (142.0)	1438.0 (29.0)	180.0 (35.3)	N/A	N/A	N/A
TSEMO	4246.3 (91.8)	2481.5 (48.5)	958.4 (49.1)	3767.4 (91.0)	1892.3 (801.5)	402.0 (31.7)
MOEAD-EGO	3474.6 (108.6)	1824.0 (40.1)	1130.3 (16.0)	4206.1 (120.5)	2526.3 (77.5)	1048.0 (37.8)

Table 4.E.1: Model fitting wall time in seconds. The mean and two standard errors of the mean are reported. All models were fit on 2x Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz. For $M = 4$, qNEHVI exceeded GPU memory during acquisition optimization and therefore has shorter average model fitting times.

Problem	Welded Beam	Vehicle Safety	Rover	Optical Design	Mazda
MORBO	7.81 (0.02)	12.58 (0.26)	9.3 (0.19)	23.57 (0.36)	172.53 (1.89)
qParEGO	0.5 (0.1)	0.1 (0.0)	51.6 (16.4)	46.7 (10.7)	N/A
TS-TCH	0.5 (0.0)	0.2 (0.0)	45.9 (1.8)	40.5 (4.9)	N/A
qNEHVI	0.5 (0.0)	0.1 (0.0)	97.8 (16.3)	46.4 (3.2)	N/A
DGEMO	N/A	N/A	809.7 (127.6)	1109.3 (178.7)	N/A
TSEMO	N/A	1.0 (0.1)	305.3 (38.2)	859.4 (131.4)	N/A
MOEA/D-EGO	N/A	0.9 (0.0)	373.2 (51.7)	736.4 (110.4)	N/A

Table 4.E.2: Model fitting wall time in seconds. The mean and two standard errors of the mean are reported. All models were fit on 2x Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz. For DGEMO, TSEMO and MOEA/D-EGO only 1,450 evaluations were performed on Rover (Trajectory Planning) and only 1,250 evaluations were performed on Optical Design, so the fitting times are shorter than if the full 2,000 evaluations had been performed.

4.F Additional Results

4.F.1 Low-dimensional problems

We consider two low-dimensional problems to allow for a comparison with existing BO baselines. The first problem we consider is a vehicle safety design problem ($d = 5$) in which we tune thicknesses of various components of an automobile frame to optimize proxy metrics for maximizing fuel efficiency, minimizing passenger trauma in a full-frontal collision, and maximizing vehicle durability. The second problem is a welded beam design problem ($d = 4$), where the goal is to minimize the cost of the beam and the deflection of the beam under the applied load [Deb and Sundar, 2006]. The design variables are the thickness and length of the welds and the height and width of the beam. In addition, there are 4 black-box constraints that must be satisfied.

Figure 4.F.1 presents results for both problems. While MORBO is not designed for such simple, low-dimensional problems, it is still competitive with other baselines such as TS-TCH and qParEGO on the vehicle design problem, though it cannot quite match the performance of qNEHVI and TSEMO.[5] The results on the welded beam problem illustrate the efficient constraint handling of MORBO.[6] On both

[5]DGEMO is not included on this problem as it consistently crashed due to an error deep in the low-level code for the graph-cutting algorithm.

[6]DGEMO, TSEMO, MOEA/D-EGO, and TS-TCH are excluded as they do not consider black-box constraints.

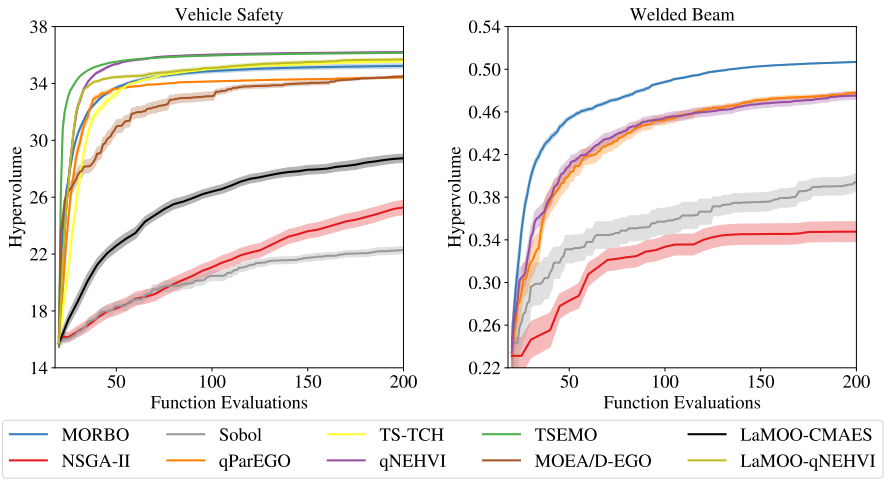

Figure 4.F.1: (Left) qNEHVI performs the best on the vehicle design problem ($d = 5$) with 3 objectives. (Right) MORBO outperforms the other methods on welded beam problem ($d = 4$) with 4 constraints.

problems, we observe that NSGA-II struggles to keep up, performing barely better (vehicle safety) or even worse (welded beam) than quasi-random Sobol exploration.

4.F.2 Candidate Generation Wall Time

Problem Batch Size	Welded Beam ($q = 1$)	Vehicle Safety ($q = 1$)	Rover ($q = 50$)	Optical Design ($q = 50$)	Mazda ($q = 50$)
MORBO	1.3 (0.0)	9.6 (0.7)	23.4 (0.4)	9.8 (0.1)	188.16 (1.72)
qParEGO	14.5 (0.3)	1.3 (0.0)	213.4 (11.2)	241.9 (14.9)	N/A
TS-TCH	N/A	0.6 (0.0)	31.3 (1.1)	48.1 (1.2)	N/A
qNEHVI	30.4 (0.4)	9.1 (0.1)	997.5 (62.8)	211.27 (6.66)	N/A
NSGA-II	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
DGEMO	N/A	N/A	697.1 (52.5)	2278.7 (199.8)	N/A
TSEMO	N/A	3.4 (0.1)	3.3 (0.0)	4.6 (0.1)	N/A
MOEA/D-EGO	N/A	44.3 (0.3)	71.1 (4.3)	97.5 (6.7)	N/A
LaMOO-CMAES	N/A	0.6 (0.0)	2.6 (0.0)	51.9 (0.3)	N/A
LAMOO-qNEHVI	N/A	24.0 (2.3)	292.4 (25.2)	258.8 (1.9)	N/A

Table 4.F.1: Batch selection wall time (excluding model fitting) in seconds. The mean and two standard errors of the mean are reported. MORBO, qParEGO, TS-TCH, and qNEHVI were run on a Tesla V100 SXM2 GPU (16GB RAM), while DGEMO, TSEMO, MOEA/D-EGO and NSGA-II were run on 2x Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz. For Welded Beam and Vehicle Safety, we ran NSGA-II with $q = 5$ in order to avoid a singleton population. For DGEMO, TSEMO and MOEA/D-EGO only 1,450 evaluations were performed on Rover (Trajectory Planning) and only 1,250 evaluations were performed on Optical Design, so the generation times are shorter than if the full 2,000 evaluations had been performed.

While candidate generation time is often a secondary concern in classic BO applications, where evaluating the black box function often takes orders of magnitude

Problem Batch Size	DTLZ3 ($M = 2$) ($q = 50$)	DTLZ5 ($M = 2$) ($q = 50$)	DTLZ7 ($M = 2$) ($q = 50$)	DTLZ3 ($M = 4$) ($q = 50$)	DTLZ5 ($M = 4$) ($q = 50$)	DTLZ7 ($M = 4$) ($q = 50$)
MORBO	26.0 (1.3)	25.1 (0.9)	293.0 (21.9)	976.9 (89.8)	973.0 (91.8)	293.0 (21.9)
qParEGO	315.8 (20.2)	299.0 (27.2)	233.0 (21.5)	372.9 (46.6)	373.1 (34.6)	232.4 (22.2)
TS-TCH	43.6 (1.4)	49.6 (2.0)	39.5 (1.9)	56.5 (1.8)	69.2 (7.5)	51.4 (3.4)
qNEHVI	2877.7 (321.3)	1879.6 (285.4)	816.9 (49.1)	4412.9 (600.7)	3778.2 (266.5)	57.6 (4.4)
NSGA-II	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.1 (0.0)	0.0 (0.0)	0.0 (0.0)
DGEMO	N/A	N/A	N/A	N/A	N/A	N/A
TSEMO	6.3 (0.1)	7.2 (0.1)	6.8 (0.1)	2878.1 (162.0)	952.0 (298.1)	22.2 (3.7)
MOEAD-EGO	277.8 (1.2)	224.9 (3.2)	245.3 (2.9)	308.7 (2.9)	303.7 (3.1)	292.2 (3.5)

Table 4.F.2: Batch selection wall time (excluding model fitting) in seconds on DTLZ problems with 2 and 4 objectives with $d = 100$. The mean and two standard errors of the mean are reported.

longer, existing methods using a single global model and standard acquisition function optimization approaches can become the bottleneck in high-throughput asynchronous evaluation settings that are common with high-dimensional problems. Tables 4.F.1 and 4.F.2 provides a comparison of the wall time for generating a batch of candidates for the different methods on the different benchmark problems. We observe that the candidate generation for MORBO is two orders of magnitudes faster than for other methods such as qParEGO and qNEHVI on the trajectory planning problem where all methods ran for the full 2,000 evaluations.

4.F.3 Pareto Frontiers

We show the Pareto frontiers for the welded beam, trajectory planning, optical design, and Mazda problems in Figure 4.F.2. In each column we show the Pareto frontiers corresponding to the worst, median, and best replications according to the final hypervolume. We exclude the vehicle design problem as it has three objectives which makes the final Pareto frontiers challenging to visualize.

Figure 4.F.2 shows that even on the low-dimensional 4D welded beam problem, MORBO is able to achieve much better coverage than the baseline methods. MORBO also explores the trade-offs better than other methods on the trajectory planning problem, where the best run by MORBO found trajectories with high reward that ended up being close to the final target location. In particular, other methods struggle to identify trajectories with large rewards while MORBO consistently find trajectories with rewards close to 5, which is the maximum possible reward. On both the optical design and Mazda problems, the Pareto frontiers

found by MORBO better explore the trade-offs between the objectives compared to NSGA-II and Sobol. We note that MORBO generally achieves good coverage of the Pareto frontier for both problems. For the optical design problem, we exclude the partial results found by running the other baselines for 1k-2k evaluations and only show the methods the ran for the full 10k evaluations. For the Mazda problem we show the Pareto frontiers of the true objectives and not the normalized objectives that are described in Section 4.6.1. MORBO is able to significantly decrease the vehicle mass at the cost of using a fewer number of common parts, a trade-off that NSGA-II fails to explore. It is worth noting that the number of common parts objective is integer-valued and that exploiting this additional information may unlock even better optimization performance of MORBO.

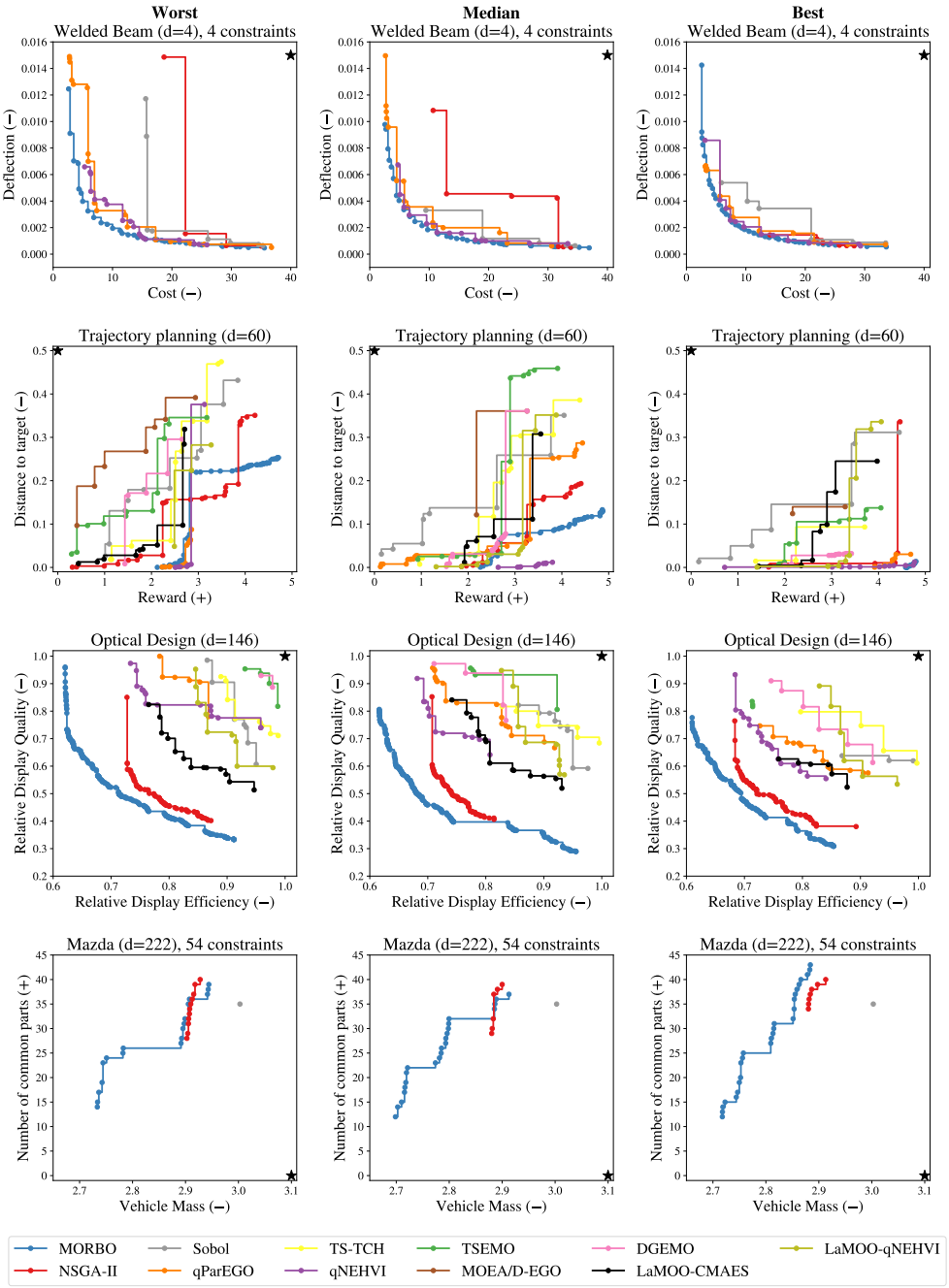

Figure 4.F.2: In each column we show the Pareto frontiers for the worst, median, and best replications according to the final hypervolume. We indicate whether an objective is minimized/maximized by $-/+$, respectively. The reference point is illustrated as a black star. The use of multiple trust regions allows MORBO to consistently achieve good coverage of the Pareto frontier, in addition to large hypervolumes.

4.F.4 Additional Benchmark Problems

To study the performance of MORBO on a broader range of problems, we evaluate MORBO on two-objective and four-objective versions of DTLZ3, DTLZ5, and DTLZ7 problems with $d = 100$. As shown in Figure 4.F.4, MORBO performs best on the four-objective DTLZ7 and achieve the best final hypervolume on the four-objective DTLZ3 problem. On the two-objective problems, MORBO always ranks in the top 4 methods as shown in Figure 4.F.4. To compare the performance in general across the DTLZ3, DTLZ5, and DTLZ7 problems with a given number of objectives, we rank the methods by the average final hypervolume across replications and compute the average rank across the three problems. As shown in Table 4.F.3, MORBO achieves the lowest rank across all methods (which is best) on both M=2 and M=4 problems. DGEMO is not evaluated on the 4-objective problems because the open-source implementation (`https://github.com/yunshengtian/DGEMO/tree/master`) does not support more than two objectives. Although DGEMO, MOEA/D-EGO and qNEHVI all perform competitively in the two objective setting, all methods are significantly slower than MORBO.

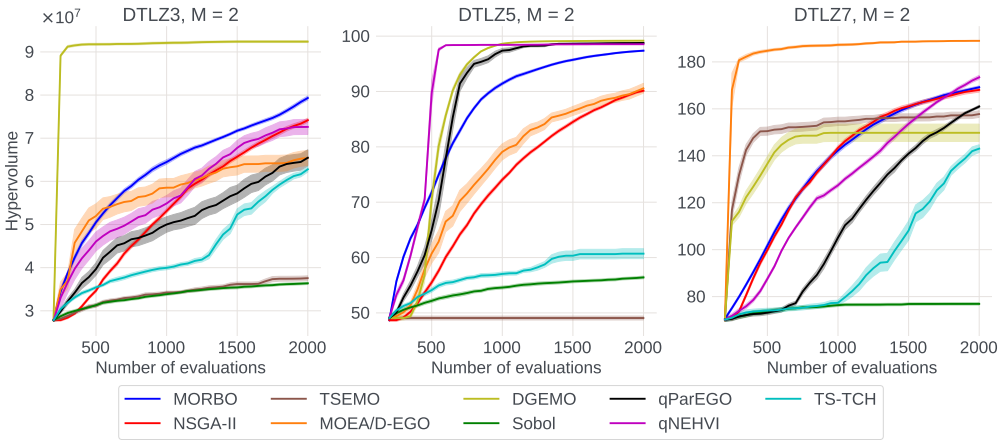

Figure 4.F.3: Optimization performance on two-objective DTLZ3, DTLZ5, and DTLZ7 problems with $d = 100$ and $q = 50$.

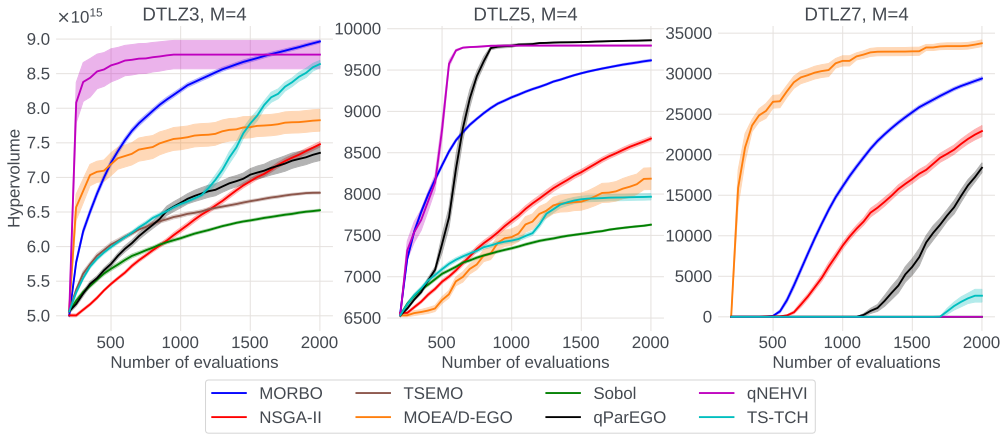

Figure 4.F.4: Optimization performance on four-objective DTLZ3, DTLZ5, and DTLZ7 problems with $d = 100$ and $q = 50$.

	Avg. Rank for M=2	Avg. Rank for M=4
MORBO	3.0	1.67
qParEGO	4.0	3.3
qNEHVI	3.0	3.16
TS-TCH	7.3	4.3
NSGA-II	4.3	3.7
DGEMO	3.0	8.2
TSEMO	7.7	8.3
MOEA/D-EGO	4.0	5.5
Sobol	8.7	6.8

Table 4.F.3: Mean rank across DTLZ3, DTLZ5, and DTLZ7 problems based on final mean hypervolume with $d = 100$ and $q = 50$. A lower rank means the method achieves better final performance on average across the DTLZ3, DTLZ5, and DTLZ7 problems with M objectives.

Endnote

Clarifications

In the introduction, we state that BO cannot easily be applied to the Mazda vehicle design and the AR/VR optical design problems. This is not an inherent limitation of BO, but rather a limitation of the commonly used surrogate models. Scalable surrogate models can be used in this setting, but poor uncertainty estimation and complex and costly training procedures limit their applicability [Li et al., 2023].

In Section 4.4.1, we describe how the batch selection procedure is fully-asynchronous. By this, we mean that a candidate design can be selected while other designs are still being evaluated by conditioning on those "pending" designs. This is different than distributed asynchronous batch selection procedures, which are not considered in this work.

In Section 4.4.2, we propose using HVC as the criterion for TR center selection because it promotes diversity. If many points are close together in output space, then the HVC of each point will be low (even if those points are very promising) and more isolated points will have higher HVC. The goal in using HVC is to improve coverage by placing the TRs in parts of the Pareto frontier with poor coverage. Nevertheless, alternative center selection techniques are an interesting direction for future work.

In Section 4.4.3, we describe how MORBO uses local modeling to improve sample efficiency by borrowing strength across TRs and how it reduces the computational load by pruning points from a TR's optimization trace that are no longer near the TR. While these two points may seem at odds, they are in fact complementary. Together, they ensure that local modeling borrows strength locally, while pruning less relevant points collected outside of the local region.

Theorem 4.5.1 provides a bound on the cumulative hypervolume regret, which is defined as $\sum_{t=1}^{T} \mathrm{HV}(\mathcal{P}^*) - \mathrm{HV}(\mathcal{P}_t)$. The proof builds on previous work on regret bounds for BO in multi-objective settings [Paria et al., 2020, Golovin and Zhang, 2020] and leverages the maximum information gain for particular GP kernels to quantify the uncertainty reduction from obtaining new observations [Srinivas et al., 2010]. The maximum information gain after T observations (which are restarts in our proof) can be bounded as $\gamma_T = O(\log(T)^{n+1})$ for the squared exponential kernel and similar bounds can be obtained for other kernels like the Matérn kernel [Srinivas et al., 2010, Golovin and Zhang, 2020].

Errata

In Lemma 4.5.1, we use $\mathbb{R}^+$ to denote the set of strictly positive real numbers.

In Theorem 4.5.1, there is a typo: the variance should be $\sigma^2_{(m)}$ not σ, which refers to the marginal variance of the m^{th} objective, which is distinct from the observation noise variance σ^2_m. In the proof of Theorem 4.5.1, "The" should not be capitalized in the first sentence. In addition, Γ refers to the Gamma function: $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ and $z > 0$ in our case.

In Section 4.6.1, "out perform" is a typo and should read "outperform". In addition, "maximize the reward" should read "maximizing the reward".

In all figures reporting performance, we plot the mean and 2 standard errors over 20 replications.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Multi-Objective Bayesian Optimization over High-Dimensional Search Spaces
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 507–517. PMLR, 01–05 Aug 2022. URL https://proceedings.mlr.press/v180/daulton22a.html.

Student Confirmation

Student Name:	Samuel Daulton		
Contribution to the Paper	I independently thought of and developed the method proposed in this paper. I have implemented the method as well as relevant baselines and spearheaded the paper writing. Together with my collaborator David, and we ran experiments to validate and analyze the proposed method.		
Signature	Samuel James Daulton	Date	28 February 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Michael Osborne, Professor of Machine Learning		
Supervisor comments	I endorse the description above, which I understand to be correct. Sam indisputably made a substantial contribution to the publication.		
Signature		Date	28 February 2023

5

Hypervolume Knowledge Gradient: A Lookahead Approach for Multi-Objective Bayesian Optimization with Partial Information

Contents

5.1 Abstract

Bayesian optimization (BO) is a popular method for sample efficient multi-objective optimization. However, existing BO techniques fail to effectively exploit common and often-neglected problem structure such as decoupled evaluations, where objectives can be queried independently from one another and each may consume different resources, and multi-fidelity evaluations, where lower fidelity-proxies of the objectives can be evaluated at lower cost. In this work, we propose a general one-step lookahead acquisition function based on the knowledge gradient that addresses the complex question of what to evaluate when and at which design points in a principled Bayesian decision theoretic fashion. By construction, our method is the one-step Bayes-optimal policy for hypervolume maximization. Empirically, we demonstrate that our method improves sample efficiency in a wide variety of real-world problems including machine learning, plasma laser acceleration, and policy optimization for content ranking. Furthermore, we show that our method

is general-purpose and yields competitive performance in standard (potentially noisy) multi-objective optimization.

5.2 Introduction

Black-box optimization is a ubiquitous problem in scientific and engineering applications. In many scenarios, there are multiple objective functions that a decision maker seeks to optimize simultaneously. Multi-objective Bayesian optimization (MOBO) is a powerful technique to achieve this with high sample efficiency [Hernandez-Lobato et al., 2016]. Most MOBO algorithms assume that all objectives are evaluated jointly (i.e. the evaluations of the objectives are *coupled*). However, in practice there are many *partial information* settings in which this is not the case. For instance, we may have the ability to evaluate objectives individually (the *decoupled evaluation* setting, see Figure 5.2.1), or we may have lower-fidelity proxies available (the *multi-fidelity* setting) in order to save time and/or resources.

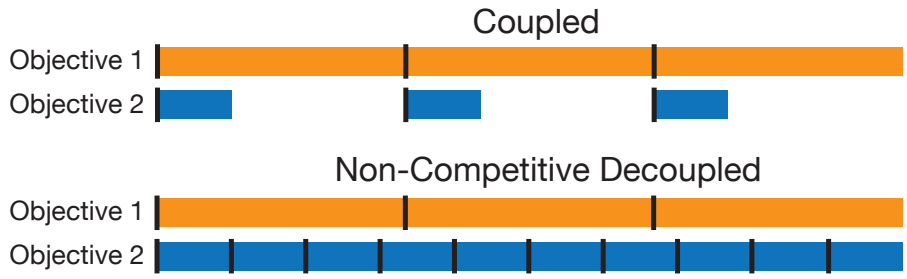

Figure 5.2.1: Decoupled evaluation allows for multiple objectives to be evaluated in non-blocking fashion (bar lengths correspond to evaluation time). With *non-competitive decoupling*, objectives have independent evaluation resources and do not compare for a shared resource.

Consider for example the problem of neural architecture search (NAS), in which we aim to identify optimal neural network architectures with respect to both model quality (e.g. accuracy) and hardware-specific metrics such as prediction latency measured on-device [Janapa Reddi et al., 2022]. In general, neither metric can be computed analytically as a function of the network architecture. Measuring accuracy typically requires a substantial amount of computation time as the NN must be trained and evaluated. Measuring latency requires access to the specific

hardware of interest (e.g. a particular mobile device type) and often few devices may be tested simultaneously [Ignatov et al., 2019]. However, device-specific latency can be evaluated on untrained NNs with reasonable accuracy with a short benchmark, making evaluation less time-consuming. This setting is illustrated in Figure 5.2.2, where we consider the scenario where we have access to a number of compute nodes—each of which can be used to train and evaluated a model—and a small number of mobile devices that can be used for measuring latency. The evaluation time for training models and evaluating accuracy will typically be much longer than the time required to measure on-device latency, but can finish asynchronously.

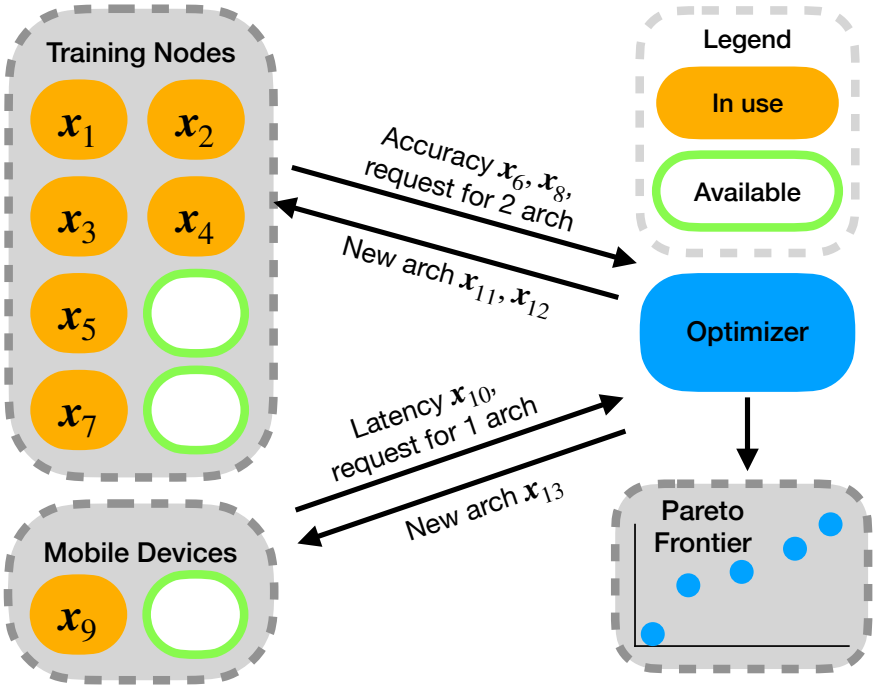

Figure 5.2.2: Multi-Objective Neural Architecture Search is one example problem with *decoupled* evaluations, where the objectives can be evaluated independently.

In this setting, a standard MOBO algorithm would simply generate architectures to be evaluated on all objectives and wait for evaluations to complete before generating new candidates (see, e.g., Guerrero-Viu et al. [2021], Eriksson et al. [2021]). This can be very inefficient, especially if evaluation time (or cost) differs substantially between the objectives. Instead, one may asynchronously choose an architecture in a *decoupled* fashion to evaluate on a given objective whenever

capacity for that objective becomes available. Figure 5.2.3 shows that even a simple policy that employs this strategy and selects architectures in a (quasi-)random fashion significantly outperforms standard MOBO methods.

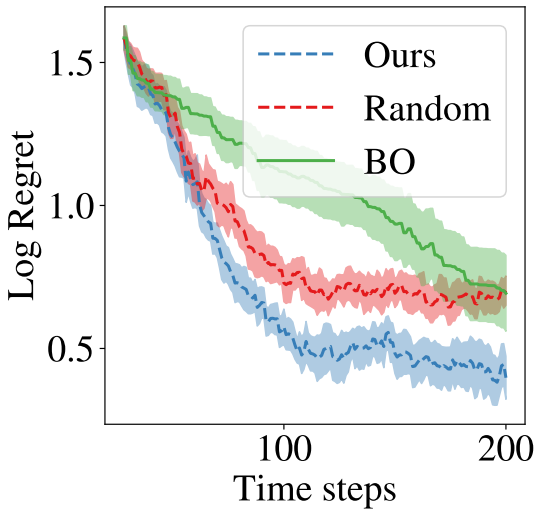

Figure 5.2.3: A random search algorithm that generates candidates for each objective in a decoupled asynchronous fashion outperforms a state-of-the-art MOBO method (qNEHVI) on a NAS problem. Our method significantly outperforms both.

Although some MOBO methods can exploit the problem's decoupled asynchronous structure (e.g., Hernandez-Lobato et al. [2016], Suzuki et al. [2020]), recent work noted that the performance improvements of existing decoupled methods relative to their non-decoupled counterparts are small [Tu et al., 2022]. In contrast, we develop a method that significantly outperforms state-of-the-art MOBO algorithms for these settings.

The NAS problem described above is an instance of a more general class of problems that is ubiquitous in the physical science and engineering. For example, in material science high-throughput screening may be applied to discover candidate compounds, but computationally-expensive simulations and/or physical experiments may be necessary to characterize the final behavior of the compound [Mukadum et al., 2021]. In the design of low-carbon-emission concrete, objectives of interest are evaluated at multiple timescales: carbon emissions can be measured within the first several hours of production, while properties such as compressive strength can take several weeks to evaluate [Barcelo et al., 2014]. Low-fidelity proxies for compressive strength (such as strength after 3 days) can be evaluated in less time, but due to the destructive nature of testing, such measurements forgo the ability

to evaluate the compressive strength at a target fidelity (e.g., after 60 days). As in the NAS example, there is limited capacity for testing certain properties (e.g., only so many rods of concrete can be cured or stored simultaneously).

The decoupled and multi-fidelity problems are instances in which a practitioner wishes to perform MOBO with *incomplete* information. By leveraging this partial information, one can reduce the cost of optimization.

Contributions

1. We formulate the MOBO problem by considering one-step look-ahead optimization of hypervolume.
2. We propose the Hypervolume Knowledge Gradient (HV-KG), a unifying acquisition strategy that allows for conditioning on incomplete information and generating candidates in a way that takes the evaluation structure into account.
3. We provide a computationally efficient technique for optimizing HV-KG and derive an unbiased gradient estimate.
4. We demonstrate substantial gains in optimization performance of HV-KG over state-of-the-art MOBO methods on a variety synthetic and real-world multi-fidelity and decoupled problems.

5.3 Preliminaries

5.3.1 Multi-Objective Optimization (MOO)

In MOO, the goal is to optimize a vector valued function $\boldsymbol{f}(x) = (f^{(1)}(x), \dots, f^{(M)}(x))$ over a compact hyperrectangular *search space* $\mathcal{X} \subset \mathbb{R}^d$. Typically there is no single best solution, and therefore the goal is to identify the set of designs with optimal objective trade-offs. We say a solution $\boldsymbol{f}(x)$ dominates another solution $\boldsymbol{f}(x')$, denoted by $\boldsymbol{f}(x) \succ \boldsymbol{f}(x')$, if $f^{(m)}(x) \geq f^{(m)}(x')$ for all m and there exists i such that $f^{(i)}(x) > f^{(i)}(x')$. An objective vector is Pareto optimal iff it is not dominated. The set $\mathcal{P}^* = \{\boldsymbol{f}(x) \mid \nexists\ x' \in \mathcal{X}\ s.t.\ \boldsymbol{f}(x') \succ \boldsymbol{f}(x)\}$ of such vectors

is called the Pareto frontier. The corresponding set of optimal *designs* is called the Pareto set $\mathcal{X}^*$ and is defined as

$$\mathcal{X}^* = \{x \in \mathcal{X} \mid \nexists\, x' \in \mathcal{X} \; s.t. \; \boldsymbol{f}(x') \succ \boldsymbol{f}(x)\}. \tag{5.1}$$

The image of $\mathcal{X}^*$ is $\mathcal{P}^*$. Given a Pareto frontier, a decision-maker can select a design with corresponding objectives that align with their preferences. The hypervolume indicator (HV) is a popular quality measure of a Pareto frontier.

Definition 5.3.1. The hypervolume indicator (HV) of a Pareto frontier $\mathcal{P}$ is the M-dimensional Lebesgue measure of the space $Z = \{z \in \mathbb{R}^M : \exists\, y \in \mathcal{P} \; s.t. \; y \succ z \succ r\}$ that is dominated by $\mathcal{P}$ and bounded from below by a reference point $r \in \mathbb{R}^M$: $\mathrm{HV}(\mathcal{P}, r) = \int_{\mathbb{R}^M} \mathbb{1}_Z(z)dz$, where $\mathbb{1}_Z(z)$ denotes characteristic function of Z.[1]

HV monotonically increases with Pareto dominance, which guarantees that it is maximized by the Pareto frontier (the image of the Pareto set) [Bader and Zitzler, 2011]:

$$\mathrm{HV}\Big[\{\boldsymbol{f}(x)\}_{x\in\mathcal{X}^*}] = \max_{\mathcal{X}'\subseteq\mathcal{X}} \mathrm{HV}[\{\boldsymbol{f}(x)\}_{x\in\mathcal{X}'}\Big]. \tag{5.2}$$

Hence we can express the goal of MOO as finding the smallest[2] set of designs $\mathcal{X}^*$ that collectively maximize the HV:

$$\mathcal{X}^* = \arg\min\Big\{|\mathcal{X}''| : \mathcal{X}'' \in \arg\max_{\mathcal{X}'\subseteq\mathcal{X}} \mathrm{HV}[\{\boldsymbol{f}(x)\}_{x\in\mathcal{X}'}]\Big\}. \tag{5.3}$$

Maximizing HV is a commonly used optimization goal that has been shown to produce high-quality approximate Pareto frontiers [Emmerich et al., 2011].

5.3.2 Bayesian Optimization (BO)

BO is a sample-efficient optimization method that models the objectives using a probabilistic *surrogate*, typically a Gaussian process (GP). Leveraging this surrogate, BO employs an *acquisition function* (AF) that quantifies the value of evaluating

[1] Henceforth, we omit r from HV for brevity.

[2] We are interested in the smallest hypervolume-maximizing set because for any hypervolume-maximizing set $\mathcal{X}' \subseteq \mathcal{X}$ and design $x \in \mathcal{X}$, $\mathrm{HV}(\mathcal{X}') = \mathrm{HV}(\mathcal{X}' \cup \{\boldsymbol{f}(x)\})$.

a new design on the objective functions. One popular AF for MOBO is expected hypervolume improvement (EHVI) [Emmerich et al., 2011], which quantifies the improvement in HV of the observed data after evaluating x:

$$\alpha_{\text{EHVI}}(x) = \mathbb{E}\Big[\text{HV}\Big(\mathcal{Y} \cup \{\boldsymbol{f}(x)\}\Big) - \text{HV}\Big(\mathcal{Y}\Big) \,|\, \mathcal{D}\Big],$$

where the expectation is over the model posterior $P(\boldsymbol{f}|\mathcal{D})$, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ are the designs evaluated so far and their corresponding observations, and $\mathcal{Y} := \{y_i\}_{i=1}^n$.

A BO policy selects one or more designs by finding the maximizer of the AF with respect to a single design x when evaluation is done sequentially, or a *batch* of designs $\boldsymbol{x} = \{x_1, \dots x_q\}$ when performing BO in parallel.[3] The designs are then evaluated on the objective functions, and the surrogate model is updated with the new observations $\boldsymbol{y}(\boldsymbol{x}) = \{y(x_1), \dots y(x_q)\}$. BO proceeds until a pre-specified evaluation budget is depleted.

5.3.3 BO with Partial Information

We briefly review terminology and common approaches to multi-fidelity (MF) BO and BO with decoupled evaluation.

Multi-fidelity BO. In multi-fidelity (MF) optimization, designs can be evaluated at different qualities within a fidelity space $\mathcal{S} \subset \mathbb{R}^K$. Examples of fidelity parameters may include the number of datapoints used to train a machine learning model or the resolution of a simulator. Lower fidelity observations are assumed to incur lower cost (e.g., compute or physical resources, time), but may differ from the value of the target objective $f^{(i)}(\,\cdot\,, s_\diamond)$, where $s_\diamond$ is a known target fidelity. MF-BO policies select designs and fidelities to query $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s})$ with the aid of a surrogate model that borrows strength across different fidelities. This can lead to significant improvements in performance within a cost budget [Poloczek et al., 2017, Takeno et al., 2020, Wu et al., 2020a, Irshad et al., 2021]. Typically, designs and fidelities are selected in a cost-aware fashion to maximize the acquisition value per unit cost. Specifically, the acquisition value of evaluating a set of designs at corresponding

[3]For the sake of generality and notational simplicity, we will assume that acquisition functions are maximized with respect a set of designs (i.e., the joint value of $\boldsymbol{x}$) throughout this work.

fidelities is weighted by the inverse of a cost function $\lambda_{\text{MF}}(\boldsymbol{x}, \boldsymbol{s}) : \mathcal{X}^q \times \mathcal{S}^q \to \mathbb{R}_{>0}$, where $\boldsymbol{s} = \{s_1, \ldots, s_q\}$.

BO with Decoupled Evaluations. In *decoupled* problems, objectives can be evaluated independently at potentially different costs. As a result, any given evaluation of a design x may not contain a full vector of outputs $y \in \mathbb{R}^M$, but rather some subset of outcomes (typically, a single objective). We further distinguish between *competitive decoupling* (CD) and *non-competitive decoupling* (NCD) [Hernández-Lobato et al., 2016]. With CD, evaluation resources are shared between objectives, whereas with NCD, they are not. Decoupled BO policies select designs to be evaluated on particular objectives. Similar to the MF setting, this is typically achieved by maximizing the acquisition value per unit cost. Here, the cost function $\lambda_{\text{D}}(\boldsymbol{x}, \boldsymbol{m}) : \mathcal{X}^q \times \mathcal{M}^q \to \mathbb{R}_{>0}$, characterizes the cost of evaluating a set of q designs, $\boldsymbol{x}$, with respect to $\boldsymbol{m} = \{m_1, \ldots, m_q\} \in \mathcal{M}^q$ objectives, where $\mathcal{M} = \{m\}_{m=1}^M$ is the set of objective indices. Similar to the MF setting, exploiting decoupling can improve optimization performance within a given budget.

5.4 Related Work

Many recent works have focused on multi-objective BO. Numerous techniques exist, the three most prominent families of methods are hypervolume-based approaches [Lukovic et al., 2020, Daulton et al., 2021, 2022b], information theoretic methods [Hernandez-Lobato et al., 2016, Belakaria et al., 2019, Suzuki et al., 2020, Tu et al., 2022, Garrido-Merchán et al., 2023], and scalarization-based techniques [Knowles, 2006, Golovin and Zhang, 2020, Daulton et al., 2022a]. However, the setting with incomplete information is much less studied.

The only methods to consider MOO with decoupled evaluations are the entropy-based Predictive Entropy Search (PESMO) [Hernandez-Lobato et al., 2016] and Pareto Frontier Entropy Search (PFES) [Suzuki et al., 2020]. Recent work on multi-objective Joint Entropy Search (JES) [Tu et al., 2022] noted that the improvements in sample efficiency appeared marginal at best in those works and therefore abstained from implementing and evaluating JES in the decoupled setting. In contrast to

this finding, we observe that exploiting decoupled evaluations with HV-KG (and even random search) can greatly improve sample efficiency.

In the MF setting, Belakaria et al. [2020] proposed MF-OSEMO, a multi-objective extension of Multi-Fidelity Max-Value Entropy Search [Takeno et al., 2020]. However, this method is only applicable in discrete fidelity settings, assumes that the objectives monotonically increase with the fidelity parameter, and, similar to Multi-Objective Max-Value Entropy Search [Belakaria et al., 2019], it suffers from significant approximation error (see Tu et al. [2022] for details). MoFiBay [Chen et al., 2022] outperforms MF-OSEMO, but is also limited to discrete fidelities. Irshad et al. [2021] introduced a MF method called MOMF, which uses the fidelity parameter as an additional "trust" objective and employs an inverse cost-weighted EHVI over all objectives. Although this approach performs quite well empirically, it does not a employ a principled procedure for selecting the fidelity parameter, and it does not specifically aim to learn the Pareto frontier over the M objectives at the target fidelity, but rather to learn the Pareto frontier over the M objectives and the trust objective. He et al. [2022] also considers a MF EHVI variant, but it is limited to the bi-fidelity setting. Guerrero-Viu et al. [2021] extend the MF BO methods BANANAS [White et al., 2021] and BOHB [Falkner et al., 2018] to the multi-objective setting, but find that full-fidelity EHVI outperforms both methods.

While our contributions build upon previous work on the Knowledge Gradient [Frazier et al., 2008, Scott et al., 2011] and its MF extensions [Poloczek et al., 2017, Wu et al., 2020a], none of these works consider the MOO setting. Q. Yahyaa et al. [2014] consider KG in the multi-objective bandit setting leveraging linear and Chebyshev scalarizations, but they do not consider the BO setting and evaluation is quite limited.

5.5 Pareto Set Selection

In MOBO, a decision maker must infer the Pareto optimal designs after receiving a finite number of observations. In the setting where observations of all objectives are available for all designs and are free of noise, a common approach is to restrict

the Pareto set selection in (5.3) to only consider dominance with respect to $X_{\mathcal{D}} := \{x : (x, \cdot) \in \mathcal{D}\}$, the set of previously evaluated designs:

$$\hat{\mathcal{X}}^* = \Big\{x \in \mathcal{X} \mid \nexists\, x' \in X_{\mathcal{D}} \; s.t. \; \boldsymbol{f}(x') \succ \boldsymbol{f}(x)\Big\}.$$

or, equivalently, $\hat{\mathcal{X}}^* = \arg\max_{\mathcal{X}' \subseteq X} \mathrm{HV}[\{\boldsymbol{f}(x)\}_{x \in \mathcal{X}'}]$. However, observations may be noisy $y \sim \mathcal{N}(\boldsymbol{f}(x), \sigma^2_{\text{noise}})$, in which case the actual objective function values are not directly observed. Similarly, in the setting where not all objectives are evaluated for all designs or not all objectives are evaluated at the target fidelity, the set of designs that have been evaluated on all objectives can be small or empty. In such scenarios, it is common for a practitioner to identify the designs that are optimal with respect to their expected values under the surrogate model [Hernandez-Lobato et al., 2016, Belakaria et al., 2019, Suzuki et al., 2020, Tu et al., 2022] and to select the optimal designs over the entire search space. Concretely, under a Bayesian decision theoretic framework, the optimal set of designs is selected as the set of designs $\mathcal{X}^*$ whose expected values under the posterior distribution of $\boldsymbol{f}$ conditional on the observed data $\mathcal{D}$ are Pareto optimal. In the standard sequential scenario,

$$\hat{\mathcal{X}}^* = \Big\{x \in \mathcal{X} \mid \nexists\, x' \in \mathcal{X} \; s.t. \; \mathbb{E}_{\mathcal{D}}[\boldsymbol{f}(x')] \succ \mathbb{E}_{\mathcal{D}}[\boldsymbol{f}(x)]\Big\},$$

where $\mathbb{E}_{\mathcal{D}}$ the expectation over the posterior of $\boldsymbol{f}$ conditional on $\mathcal{D}$. An equivalent problem is to find the set of designs that maximize the HV of the expected values:

$$\hat{\mathcal{X}}^* = \arg\max_{\mathcal{X}' \subseteq \mathcal{X}} \mathrm{HV}\Big[\Big\{\mathbb{E}_{\mathcal{D}}\Big[\boldsymbol{f}(x)\Big]\Big\}_{x \in \mathcal{X}'}\Big]. \tag{5.4}$$

Since $\hat{\mathcal{X}}^*$ can be an infinite set, it is typically hard to identify exactly. A common approach is to identify a finite-cardinality approximate Pareto set $\hat{X}^*$ containing N_p designs, typically by running an evolutionary algorithm such as NSGA-II on $\mathbb{E}_{\mathcal{D}}[\boldsymbol{f}(x)]$ [Hernandez-Lobato et al., 2016, Belakaria et al., 2019, Suzuki et al., 2020, Tu et al., 2022]. Often, the HV of the resulting Pareto frontiers is used for comparing their quality. We can directly express this optimization goal by restricting the HV maximization problem in Equation (5.4) to finite cardinality sets $|\mathcal{X}'| \leq N_p$:

$$\hat{X}^* = \arg\max_{X \subseteq \mathcal{X},\; |X| \leq N_p} \mathrm{HV}\Big[\Big\{\mathbb{E}_{\mathcal{D}}\Big[\boldsymbol{f}(x)\Big]\Big\}_{x \in \mathcal{X}'}\Big].$$

Henceforth, we assume $|X| \leq N_p$. In the following, we will write $\boldsymbol{\mu}(X \mid \mathcal{D}) := \{\mathbb{E}_{\mathcal{D}}[\boldsymbol{f}(x)]\}_{x \in X}$.

5.6 A Knowledge Gradient Approach

Given the Bayesian decision theoretic goal above, we derive a novel AF to explicitly target our end goal: inferring a hypervolume-maximizing finite Pareto set. Consider the scenario where one can obtain additional observations $(\boldsymbol{x}, \boldsymbol{y})$ before identifying the Pareto optimal designs conditional on $\mathcal{D}_{\boldsymbol{x}} := \mathcal{D} \cup \{(\boldsymbol{x}, \boldsymbol{y})\}$. Then, the one-step Bayes-optimal acquisition function, denoted as the Hypervolume Knowledge Gradient (HV-KG), is:

$$\alpha_{\text{HV-KG}}(\boldsymbol{x}) = \mathbb{E}_{\mathcal{D}}\left[\max_{X \subseteq \mathcal{X}} \text{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big] - \psi^*\right], \tag{5.5}$$

where $\psi^* := \max_{X \subseteq \mathcal{X}} \text{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D})\Big]$. Conceptually, HV-KG quantifies the increase in hypervolume of the Pareto frontier across the expected values of the objectives. The outer expectation is necessary because $\boldsymbol{y}$ is a random variable and $\boldsymbol{\mu}$ depends on $\boldsymbol{y}$. Since ψ^* is constant conditional on $\mathcal{D}$, the maximizer of $\alpha_{\text{HV-KG}}(\boldsymbol{x})$ does not change if ψ^* is omitted.

Asynchronous Candidate Generation While (5.5) is the formulation for *parallel* (i.e. batch) candidate generation, it is straightforwardly extended to the setting of *asynchronous* generation, in which the result of some *pending points* $\tilde{\boldsymbol{x}}$ have yet to be observed. In this case the acquisition function is evaluated on $\boldsymbol{x} \cup \tilde{\boldsymbol{x}}$ but optimized only over $\boldsymbol{x}$.

5.7 Conditioning on Partial Information

Although HV-KG is applicable to standard MOBO problems where observations of all objectives are received for all designs, a key benefit of HV-KG is that it enables conditioning on incomplete information. In contrast, other popular HV-based methods (e.g. Emmerich and Fonseca [2011], Lukovic et al. [2020], Daulton et al. [2021]) cannot condition on incomplete information because the

rely on utility functions that measure improvement with respect an in-sample Pareto set and assume that observations of all objectives will be received for the selected candidate design. In contrast, HV-KG can leverage incomplete information (such as decoupled and multi-fidelity evaluations) simply by changing the new data $\mathcal{D}_{\boldsymbol{x}}$ that the model is conditioned on. Note that in this section we use the notation specified in Section 5.3.3.

Decoupled Evaluations In the decoupled setting the objectives can be evaluated independently. The decoupled HV-KG acquisition function is[4]

$$\alpha_{\text{D-HV-KG}}(\boldsymbol{x}, \boldsymbol{m}) = \frac{\alpha_{\text{HV-KG}}(\boldsymbol{x})}{\lambda_D(\boldsymbol{x}, \boldsymbol{m})},$$

where now $\mathcal{D}_{\boldsymbol{x}} = \{(x_i, y_i^{(m_i)})\}_{i=1}^q$. In CD, the evaluation budget is in terms of total cost and all objectives compete for shared resources. As such, we consider the $q = 1$ case, without loss of generality. The BO policy chooses the objective m and design x jointly in a cost-aware fashion. In NCD, the evaluation budget is in terms of time and all available evaluation capacity should be exploited.

Let $c \in \mathbb{N}^M$ denote the available evaluation capacity for each objective. The policy generates $q = \sum_{m=1}^M c^{(m)}$ candidates $\boldsymbol{x}$ jointly to exploit all available capacity. Each candidate is assigned to be evaluated on an objective specified by $\boldsymbol{m} \in \mathcal{M}^q$ such that $c^{(m)} = \sum_{i=1}^q \mathbb{1}(m_i = m)$ for all $m = 1, ..., M$.

Multi-Fidelity Let $\boldsymbol{\mu}_\diamond(X, \mathcal{D}) := \{\mathbb{E}_{\mathcal{D}}[\boldsymbol{f}(x, \boldsymbol{s}_\diamond)]\}_{x \in X}$. The multi-fidelity HV-KG AF is given by[4]

$$\alpha_{\text{MF-HV-KG}}(\boldsymbol{x}, \boldsymbol{s}) = \frac{1}{\lambda_{\text{MF}}(\boldsymbol{x}, \boldsymbol{s})} \mathbb{E}_{\mathcal{D}}\left[\max_{X \subseteq \mathcal{X}} \text{HV}\left[\boldsymbol{\mu}_\diamond\left(X \mid \mathcal{D}_{(\boldsymbol{x}, \boldsymbol{s})}\right)\right] - \psi_\diamond^*\right],$$

where $\psi_\diamond^* := \max_{X \subseteq \mathcal{X}} \text{HV}\left[\boldsymbol{\mu}_\diamond(X \mid \mathcal{D})\right]$ and $\mathcal{D}_{(\boldsymbol{x}, \boldsymbol{s})} := \mathcal{D} \cup \{(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{y})\}$. We note that in this general MF-HV-KG formulation, each objective has a (potentially empty) set of fidelity parameters, which can contain (i) fidelity parameters that are unique

[4]We clamp the difference in HV inside the expectation in (5.5) to ensure the numerator remains non-negative in the cost-weighted variants. See Appendix 5.C for discussion.

to that objective (e.g. and (ii) fidelity parameters that are shared amongst multiple objectives, or (iii) a combination of (i) and (ii).

5.8 Computing and Optimizing HV-KG

5.8.1 Unbiased Estimation

Although HV-KG cannot be computed analytically, we obtain an unbiased estimator by approximating the outer expectation via Monte Carlo:

$$\hat{\alpha}_{\text{HV-KG}}(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\left(\max_{X_i \subseteq \mathcal{X}} \text{HV}\Big[\boldsymbol{\mu}(X_i \mid \mathcal{D}^i_{\boldsymbol{x}})\Big]\right) - \psi^*, \tag{5.6}$$

where $\mathcal{D}^i_{\boldsymbol{x}} = \mathcal{D} \cup \{\boldsymbol{x}, \boldsymbol{y}^i\}$ with each $\boldsymbol{y}^i$ a realization or "fantasy" sample of the random variable $\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})$. For each fantasy $\boldsymbol{y}^i$, the updated posterior mean can be computed analytically [Frazier et al., 2008]. The inner maximization involves a numerical optimization over a $(N_p \cdot d)$-dimensional space conditional upon the selected $\boldsymbol{x}$.

5.8.2 Hypervolume Computation

To enable efficient optimization, one would like to compute the hypervolume in a differentiable fashion. The joint HV of N_p points can be computed exactly using the inclusion exclusion principle (IEP) [Lopez et al., 2015] and this approach is differentiable with respect to X_i [Daulton et al., 2020]. The IEP scales exponentially with N_p and therefore is only be feasible for small N_p, but a small N_p tends to work empirically here and for information theoretic approaches. Following Tu et al. [2022], we select $N_p = 10$ (and find that HV-KG is robust to choice of N_p in Appendix 5.D.3).

5.8.3 Nested Optimization

A common approach for solving the nested optimization problems in KG methods is to leverage the envelope theorem to obtain an unbiased gradient estimator [Wu et al., 2017, 2020a] and solve the inner optimization to completion whenever $\boldsymbol{x}$ changes.

We derive a gradient of HV-KG in Theorem 5.B.1 in Appendix 5.B.2, which can be estimated without bias via Monte Carlo and optimized via stochastic gradient ascent.

However, solving the inner optimization problem to completion after each outer optimization step is computationally intensive and impractically slow (see Figure 5.D.6).

5.8.4 Deterministic Estimation and Optimization

Instead, we opt for using sample-average (SAA) approximation as Balandat et al. [2020]. Using a fixed set of the standard normal *base samples* $\boldsymbol{\epsilon} := \{\epsilon^i\}_{i=1}^N$, $\epsilon^i \in \mathbb{R}^M$ for the fantasized observations $y^{i,(m)} = \mu_{\mathcal{D}}^{(m)}(\boldsymbol{x}) + L_{\mathcal{D}}^{(m)}(\boldsymbol{x})\epsilon^{i,(m)}$, where $L_{\mathcal{D}}^{(m)}$ is the Cholesky factor of the posterior covariance matrix, the fantasies y^i and updated posterior mean functions $\boldsymbol{\mu}(\cdot\,|\mathcal{D}_{\boldsymbol{x}}^i)$ are deterministic (see Appendix 5.B.1). Given fixed base samples, we can interchange maximization and summation in (5.6) to obtain

$$\hat{\alpha}_{\text{HV-KG}}(\boldsymbol{x}) = \max_{X_1,\dots,X_N \subseteq \mathcal{X}} \frac{1}{N}\sum_{i=1}^{N} \text{HV}\Big[\boldsymbol{\mu}(X_i \,|\, \mathcal{D}_{\boldsymbol{x}}^i)\Big] - \psi_t^*. \tag{5.7}$$

The SAA estimator in (5.7) can be maximized efficiently by optimizing over $\{\boldsymbol{x}, X_1, \dots, X_N\}$ simultaneously in "one shot" [Balandat et al., 2020]. Although such an approach requires optimizing over a $((N_p \cdot N + 1) \cdot d)$-dimensional space, HV-KG is differentiable with respect to $\boldsymbol{x}, X_1, \dots, X_N$ and sample-path gradients can be computed via auto-differentiation. Since the SAA estimator is deterministic, (quasi-) second-order gradient-based optimizers can be employed. We can show that the maximizer $\boldsymbol{x}_N^*$ of our SAA estimator converges with probability one to an element of $\mathcal{X}_{\text{HV-KG}}^* = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha_{\text{HV-KG}}(\boldsymbol{x})$, the set of optimizers of the true $\alpha_{\text{HV-KG}}$, and that convergence occurs exponentially fast in the number of MC samples N.

Theorem 5.8.1. *Suppose that $\mathcal{X}$ is compact and that $\boldsymbol{f} \sim GP(\mu_0(\cdot), K_0(\cdot,\cdot))$ is a sample from a multi-output Gaussian process prior with continuously differentiable mean $\mu_0(\cdot)$ and covariance $K_0(\cdot,\cdot)$ functions. Let $\{\epsilon_i\}_{i=1}^N$ be i.i.d. base samples from $\mathcal{N}(0, I_M)$ and let $\boldsymbol{x}_N^* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \hat{\alpha}_{HV\text{-}KG}^N(\boldsymbol{x})$, then*

(i) $\hat{\alpha}_{HV\text{-}KG}(\boldsymbol{x}_N^*) \to \alpha_{HV\text{-}KG}^*$ a.s.

(ii) $\inf_{\boldsymbol{x}^* \in \mathcal{X}^*_{\text{HV-KG}}} ||\boldsymbol{x}^*_N - \boldsymbol{x}^*|| \to 0$ *a.s.*

(iii) $\forall\ \delta > 0, \exists\ K < \infty, \alpha > 0$ *such that*

$$p\Big(\inf_{\boldsymbol{x}^* \in \mathcal{X}^*_{\text{HV-KG}}} ||\boldsymbol{x}^*_N - \boldsymbol{x}^*|| \geq \delta\Big) \leq Ke^{-\alpha N}.$$

We find that optimizing Equation (5.7) using L-BFGS-B yields strong performance using the initialization technique described in Appendix 5.A.2.

5.9 Experiments

We evaluate HV-KG on synthetic and real-world problems including multi-fidelity problems and problems with decoupled evaluations. For all HV-KG variants, we use $N = 32$ fantasies and $N_p = 10$. Because HV-KG is the only acquisition function that handles all cases, we consider differing methods for different types of partial information. As a baseline, we include qNEHVI due to its consistent performance in all our tests, and scrambled Sobol sequences [Owen, 1998] as a quasi-random baseline. In the multi-fidelity case, we include a comparison with MOMF. For decoupled sampling, we compare with two information-based AFs, JES and PFES. JES has been shown at least as well as all other ES-based methods [Tu et al., 2022] and can straightforwardly be generalized to the decoupled setting (although it was not evaluated in the original paper). PFES has a decoupled variant that had not been evaluated outside of [Suzuki et al., 2020]. All AFs are implemented in BoTorch and utilize GPs with a standard Matérn 5/2 kernel over the design space (see Appendix 5.A for additional details).

To compare methods, we first solve (5.4) by optimizing the posterior means using NSGA-II [Deb et al., 2002] to find the model-estimated Pareto set. Then, we compute the true objective values for the designs in the model-estimated Pareto set and compute the resulting hypervolume dominated by the true Pareto frontier of the model-selected Pareto set. This procedure is common in many works (cf., [Hernandez-Lobato et al., 2016, Belakaria et al., 2019, Suzuki et al., 2020, Tu et al., 2022]). We report means and 2 standard errors of the mean across 20 replications of the log hypervolume regret: the difference in hypervolume between the image of Pareto set identified by the method and the true Pareto frontier.

While the focus of this work is on MOBO with partial information, we also include an evaluation of all applicable methods for the standard noiseless and noisy case with complete information in Appendix 5.D. We find that HV-KG performs at least as well as other methods in all test problems considered. Additional details about test problems in the remainder of this section can be found in Appendix 5.A.

5.9.1 Multi-Fidelity

We consider the performance of HV-KG relative to MOMF, a MF MOBO method, as well as the other non-MF baselines with respect to four MF test problems.

Synthetic Problems *(1)* Park ($d = 4$ inputs, $M = 2$ objectives) *(2)* MF Branin Currin ($d = 2, M = 2$) where the cost function is $\lambda(s) = \exp(4.8s)$ [Irshad et al., 2021].

Real-World Problems We consider two problems to highlight the importance of exploiting multi-fidelity information sources: *(1)* **Laser-plasma acceleration**, ($d = 4, M = 3$) from Irshad et al. [2023a], where a continuous fidelity parameter governs the simulation accuracy and simulation time. *(2)* **Recommender system ranking policy optimization** ($d = 15, M = 2$) from Liu et al. [2023] simulates a ranking policy which controls the number of items retrieved from different content sources in a recommender system. The target objectives are long-term engagement with the product and content serving cost, and the fidelity parameter is the experiment duration. This problem is designed to mimic setups common to Bayesian optimization of ranking policies with "A/B tests" [Letham and Bakshy, 2019], where selection bias and transient effects bias objectives in the short term [Bakshy et al., 2014].

5.9.2 Decoupled Evaluation

In the decoupled setting, we compare against three decoupled methods: decoupled PFES [Suzuki et al., 2020], the decoupled extension of JES-LB2 proposed in Tu et al. [2022] (see Appendix M), and a decoupled variant of Sobol. For Sobol, evaluated

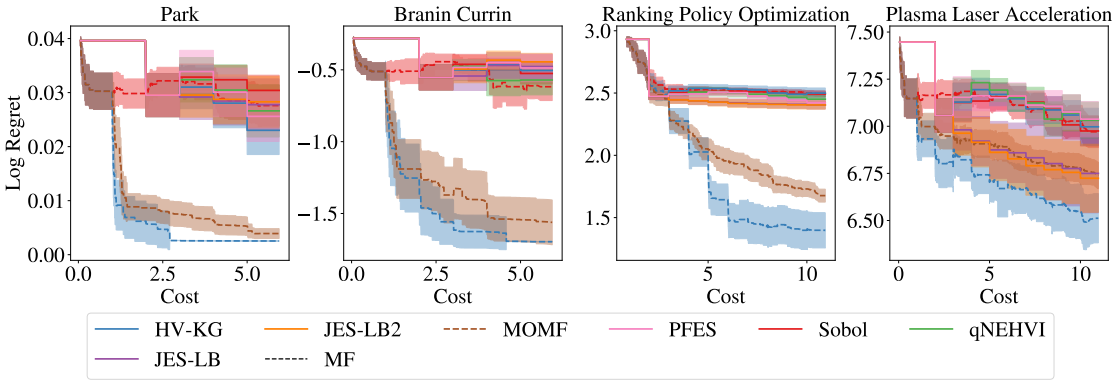

Figure 5.9.1: Multi-fidelity optimization performance.

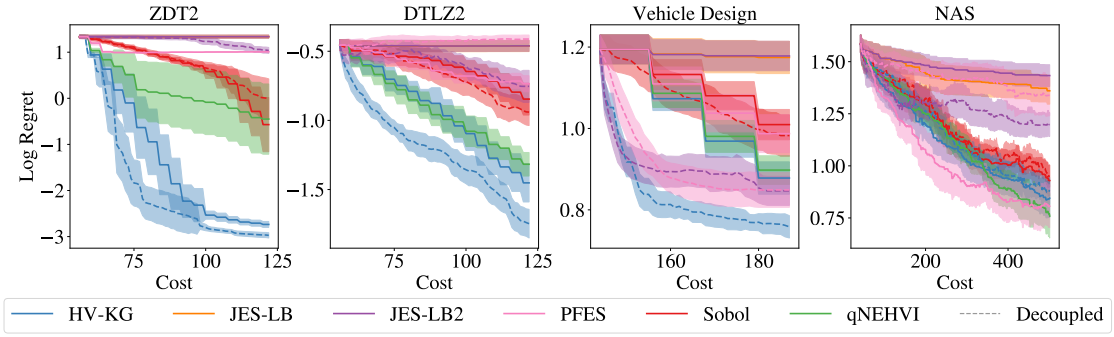

Figure 5.9.2: Optimization performance with *competitive decoupling*.

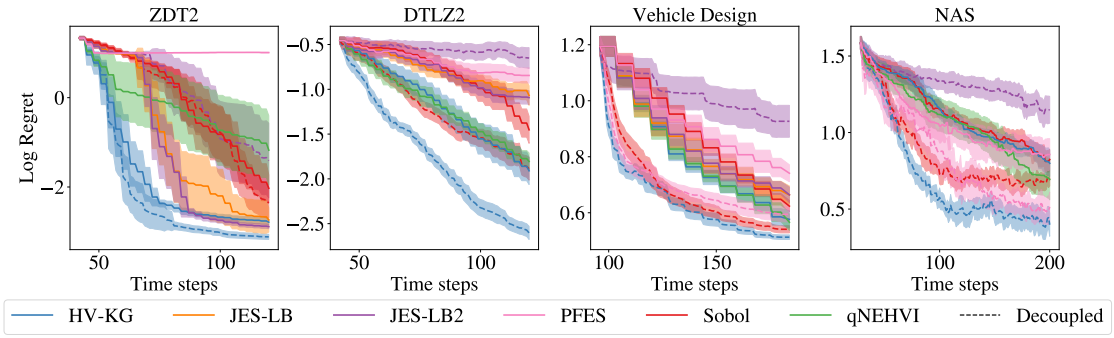

Figure 5.9.3: Optimization performance with *non-competitive decoupling*.

objectives are selected uniformly at random, and designs are sampled via scrambled Sobol sequences. We consider both types of coupling, CD, where evaluations occur sequentially, and NCD, where evaluations occur asynchronously.

Synthetic Problems We evaluate performance on the classic ZDT2 ($d = 6, M = 2$) and [Zitzler et al., 2000] and DTLZ2 ($d = 6, M = 2$) [Deb et al., 2002] test problems, and evaluate the objectives in a decoupled fashion. For CD, ZDT1 and

DLTZ2 use a cost ratio of 1:3, and for NCD, the objectives have a evaluation time ratio of 1:3, each has a capacity of 1 and equal cost (here we are only concerned with time for NCD).

Real-World Problems We consider two real-world problems: (1) **NAS** ($d = 6, M = 2$) is the neural architecture search problem we use to motivate non-competitive decoupling in the Section 5.2. The goal is to maximize accuracy and minimize on-device latency for an ImageNet model. Here, we use data from NASBench201 [Dong and Yang, 2020] and HW-NAS-Bench [Li et al., 2021] for the first and second objective, respectively. The NCD version of the problem depicted in Figure 5.2.2. The training and latency objectives have an evaluation time ratio of 1:4 and capacities of 2 and 8 respectively and equal cost. For CD, latency and accuracy have costs 1 and 2, respectively. (2) **Vehicle Design** ($d = 5, M = 3$) poses an automotive design problem, where the goal is optimize the design a vehicle to maximize fuel economy, minimize vehicle damage in an off-frontal collision, and minimize passenger trauma in a full frontal crash [Liao et al., 2008]. We leverage the surrogate from Tanabe and Ishibuchi [2020] for this problem. For CD, the three objectives have a cost ratio of 1:3:8, and for NCD, the objectives have an evaluation time ratio of 1:3:8 and each objective has an evaluation capacity of 1 and equal cost.

5.9.3 Results

We find that MF-HV-KG and decoupled HV-KG variants significantly improve sample efficiency and optimization performance compared to alternatives. Notably, Sobol baselines that exploit problem structure perform remarkably well on many problems. As shown in Figure 5.9.1, MOMF performs well on many MF problems, but is never better than MF-HV-KG. In the decoupled setting, the entropy-based decoupled methods struggle on many tasks and we find them to be sensitive to the cost function, whereas HV-KG is more robust across problems and costs (see Appendix 5.D). With CD, HV-KG is again the top performer on 3/4 problems with the exception of NAS as shown in Figure 5.9.2. The poor performance of

all methods on the NAS CD problem is likely due to poor surrogate model fits; selecting which objective to query in cost-aware fashion depends on having a well-specified model. On the other hand, in NCD, decoupled methods do not need to rely on the model to select which objective to evaluate and can simply utilize all evaluation capacity. Unsurprisingly, non-decoupled methods perform poorly because they can only generate candidates once all metrics have been evaluated and are limited to the lowest evaluation capacity across all outcomes. Finally, we find (in Appendix 5.D.4) that HV-KG generates candidates faster than entropy-based methods in the decoupled setting.

5.10 Discussion

HV-KG provides a principled approach to multi-objective Bayesian optimization with incomplete information, including situations in which objective values may be queried separately or at multiple fidelities. To the best of our knowledge, this is the first paper to consider a KG-based approach in the decoupled setting and the multi-objective setting with partial information, and we show that we are able to obtain state-of-the-art performance with respect to standard, decoupled, and multi-fidelity MOBO.

Our work opens the door to exploiting other problems with incomplete observations. Although we exploit MF evaluations, it is possible to leverage more sophisticated models that consider the evolution of objectives over time by leveraging trace observations such as learning curves in AutoML or reinforcement learning [Wu et al., 2020a, Nguyen et al., 2020b]. The HV-KG approach also lends itself to other instances where incomplete data is available. For example, in contextual BO, we may wish to identify the best configuration or set of best configurations across all contexts, and can transfer knowledge from one context to another. For instance, in the context of on-device AI one may target multiple possible devices and when developing green concrete, one may wish to develop mixes that are efficient across a variety of environments (e.g., temperature conditions). In other cases, experiments may involve multiple dependent stages, such as cascade and function networks [Astudillo and Frazier, 2021, Kusakawa et al., 2022] common in

manufacturing pipelines. Extensions of HV-KG could be used to target learning at various stages. In addition, we leave exploring alternative ways of handling discrete parameters in HV-KG (e.g. Moss et al. [2020], Jain et al. [2022] to future work. Finally, while the performance of HV-KG is competitive with other methods, the speed of these algorithms can be further improved via alternative approaches to computing HV [Shang et al., 2022].

Appendix

5.A Experiment Details

5.A.1 Implementation of Acquisition Functions and Models

We use the BoTorch implementations of qNEHVI [Daulton et al., 2020], MOMF [Irshad et al., 2021], JES-LB and JES-LB2 [Tu et al., 2022] developed by the original authors of these works.[5] To determine the Pareto frontier, we use Tu et al. [2022]'s NSGA-II-based implementation.[6] For PFES [Suzuki et al., 2020], we utilize Tu et al. [2022]'s open source implementation in BoTorch, which includes the lower-bound batch variant (for $q > 1$). For decoupled sampling, we modified the existing BoTorch implementation of Suzuki et al. [2020] to include the decoupled approach from the original paper, and we implemented the extension of JES-LB2 to the decoupled setting, proposed in Tu et al. [2022, Appendix M]. Additional details about all implementations can be found in the code included in the supplementary material.

For all MC acquisition functions, we use quasi-random (QMC) base samples and sample average approximation [Balandat et al., 2020]. All methods are optimized using L-BFGS-B from 20 starting points using the default initialization heuristic in BoTorch [Balandat et al., 2020]—except for HV-KG, which we optimize from a single starting point to limit computational overhead. All methods use independent GPs with ARD Matérn 5/2 kernels. We use Gamma(2,2) priors over the lengthscales (with allow learning large lengthscales for irrelevant parameters) and Gamma(2, 0.15) priors on the outputscales. We assume that the noise level is known.

For the NAS and chemistry problems, we one-hot encode each categorical x with C categories as $\boldsymbol{x}' = [x'_1, ..., x'_C] \in [0, 1]^C$, apply exact discretization functions

[5] Code is available in open source at `https://github.com/pytorch/botorch`.
[6] Code is available in open source at `https://github.com/benmltu/JES`.

(i.e., $x = \text{ONE-HOT}(\arg\max_{c \in C} x'_c)$) before evaluating the GP, and use straight through-gradient gradient estimators [Daulton et al., 2022c].

5.A.2 Initialization of HV-KG

The optimization of HV-KG can be significantly sped up by choosing good initial conditions for the design point and the fantasy optimizers. If we assume that the additional observation $(\boldsymbol{x}, \boldsymbol{y})$ does not drastically change the location of the Pareto set in input space, then solving the optimization problem

$$\max_{X \subseteq \mathcal{X}} \text{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D})\Big] \tag{5.8}$$

under the current posterior (having observed data $\mathcal{D}$) will yield an optimizer that will likely be quite close to the optimal $\{X_i\}_{i=1}^N$ after fantasizing about the unknown function values $\boldsymbol{y}$. Equation (5.8) can be solved efficiently using gradient-based optimization. Using different starting points, we can identify N solution sets X to use as initial values for the HV-KG optimization problem. Lastly, we can find starting points for $\boldsymbol{x}$ conditional on $(X_1, \ldots, X_N)$ using standard BO initialization heuristics such as Boltzmann sampling on the HV-KG values [Balandat et al., 2020]. We use the resulting starting point $(\boldsymbol{x}, X_1, \ldots, X_N)$ to optimize HV-KG via quasi-second order methods (L-BFFS-B) using SAA.[7]

5.A.3 Problem Details

All noisy variants use additive zero-mean Gaussian noise, where the noise standard deviations (denoted by σ) are set as a percentage of the range of each objective as indicated in parentheses. These noise levels come from in previous works [Daulton et al., 2021, Tu et al., 2022].

We use the multi-fidelity versions of Park ($M = 2$ objectives, $d = 4$ inputs, $K = 1$ fidelity parameters) and Branin-Currin ($M = 2, d = 2, K = 1$) from Irshad

[7] In the case that the objectives are not modeled directly and do not have analytic expressions in terms of the model outputs, the inner expectation could also be approximated with Monte Carlo samples. Such cases would arise in the case of constrained optimization where the goal is to optimize feasibility-weighted objectives and unweighted objectives and constraint slacks are both modeled and subsequently combined.

et al. [2021], the Penicillin manufacturing problem ($M = 3, d = 7, \sigma = 1\%$) from Liang and Lai [2021].[8] When considering the standard test problems DTLZ2 ($M = 2, d = 6, \sigma = 10\%$) [Deb et al., 2002], ZDT2 ($M = 2, d = 6, \sigma = 10\%$) [Zitzler et al., 2000], and Vehicle Design ($M = 3, d = 5, \sigma = 1\%$) [Tanabe and Ishibuchi, 2020] all of which are implemented in BoTorch.[0]

We use implementations of the **Marine** design problem ($M = 4, d = 6, \sigma = 3\%$) [Parsons and Scott, 2004, Tanabe and Ishibuchi, 2020] and **SnAr** ($M = 2, d = 4\sigma = 3\%$, a chemical reaction optimization problem) [Hone et al., 2017] from Tu et al. [2022].[0]

Chemistry problem aims to tune experimental conditions to maximize chemical reaction yield while minimizing cost ($M = 2, d = 5$). We adopt this problem from Daulton et al. [2022c] (*Direct Arylation Chemical Synthesis*). A GP surrogate is fit to chemical reaction data from Shields et al. [2021],[9] and corresponding reaction cost data[10] from Torres et al. [2022]

NAS problem ($M = 2, d = 6$**)**, we use accuracy data from NASBench201[11] [Dong and Yang, 2020], augmented with edge GPU latency estimates from HW-NAS-Bench[12] [Li et al., 2021].

Vehicle Design problem ($d = 5, M = 3$) poses a hypothetical automotive problem. We leverage the surrogate from Tanabe and Ishibuchi [2020] and formulate the problem the problem with respect to the surrogate in the following way: we minimize mass (a proxy for maximizing fuel economy), minimize length of toe-box intrusion in case of a crash (a proxy for vehicle damage), and minimize acceleration (a proxy for passenger trauma), vehicle damage in an off-frontal collision (measure in a by toe-box intrusion distance), and minimize acceleration (a proxy for passenger trauma in a full frontal crash) [Liao et al., 2008]. This problem can most naturally

[8] We modify the search space slightly to make the simulations less bimodal (as identified in Park et al. [2022]) by reducing the number of designs that lead to a zero fermentation time objective. The modified search space sets the lower bounds of the 4^{th} and 5^{th} parameters to be 4 and $\frac{1}{4}$, respectively.

[9] Data is available at `https://github.com/b-shields/edbo`.

[10] Data is available at `https://github.com/doyle-lab-ucla/edboplus`.

[11] Code is available at `https://github.com/D-X-Y/NAS-Bench-201`.

[12] Code is available at `https://github.com/GATECH-EIC/HW-NAS-Bench/`.

Table 5.A.1: Reference points for all benchmark problems (assuming minimization of all objectives). In our benchmarks, we maximize all objectives by multiplying objectives and reference points by -1.

Problem	Reference Point
ZDT2	(11, 11)
DTLZ2	(1.1,1.1)
Vehicle Design	(1698.55, 11.21, 0.29)
NAS	(-7.319, 30.847)
Park	(0,0)
Branin-Currin	(0,0)
Ranking Policy Optimization	(5.353, -44.39)
Plasma Laser Acceleration	(280.864, -50.613, -36.412)
Penicillin	(-5.657, 64.1, 340.0)
Marine	(-250, $2 \cdot 10^4$, $2.5 \cdot 10^4$, 15)
SnAr	(-5.5, 5)
Chemistry	(32.669, -0.107)

be thought of as a NCD problem, since the evaluation of the last two objectives as destructive, so that each objective requires a different type of collision. The fuel economy objective is less costly to evaluate, as it does not require manufacturing and crashing a car. For CD, the three objectives have a cost ratio of 1:3:8, and for NCD, the objectives have an evaluation time ratio of 1:3:8 and each objective has an evaluation capacity of 1 and equal cost.

We use variant of the ranking policy ($M = 2, d = 15, K = 1$) optimization problem system from [Liu et al., 2023]. To create a multi-fidelity variant of this problem, we add bias term to emulate the "novelty effect", an ephemeral boost in engagement, that commonly affects engagement metrics when new ranking policies conducted via "A/B tests" [Bakshy et al., 2014]. Running longer experiments (high fidelity experiments), will reduce the novelty effect and provide more accurate estimates of the long term effect. We use the same search space as in Liu et al. [2023], but restricted to 15 dimensions.

The plasma laser acceleration problem comes from the recent work by Irshad et al. [2023a]. We fit GP surrogate models to the data [Irshad et al., 2023b] collected by the original authors via simulations.[13]

[13]Data is available at `https://doi.org/10.5281/zenodo.7565882`.

5.A.4 Initial Point Selection for Multi-Fidelity Experiments

The cost budget for selecting initial design points is set equal to the cost of 2 full-fidelity evaluations $2\lambda(s)$. Full fidelity methods sample 2 designs from a scrambled Sobol sequence. Multi-fidelity methods sample the design parameters uniformly at random and the fidelity parameter is sampled (via the inverse transform) from the probability distribution with pdf $p(s) \propto \frac{1}{\lambda(s)}$. Designs are added until the next sampled point exceeds the cost budget. Hence, multi-fidelity methods use an initialization with cost $\leq 2\lambda(s)$.

5.B Theoretical Results

5.B.1 Preliminaries

Hypervolume Computation

For a set of N_p points $\mathcal{Y} = \{y_j\}_{j=1}^{N_p}$, the HV w.r.t to a reference point r can be computed in a differentiable fashion [Daulton et al., 2020] as

$$\mathrm{HV}(\mathcal{Y}, r) = \sum_{j=1}^{N_p} \sum_{Y_j \in \mathcal{Y}_j} (-1)^{j+1} \prod_{m=1}^{M} \left[z_{Y_j}^{(m)} - r^{(m)} \right]_+, \tag{5.9}$$

where $\mathcal{Y}_j := \{Y_j \subseteq \mathcal{Y} : |Y_j| = j\}$ is the set of all subsets of $\mathcal{Y}$ of size j and $z_{Y_j}^{(m)} := \min \left[y_{i_1}^{(m)}, \ldots, y_{i_j}^{(m)} \right]$ for $Y_j = \{y_{i_1}, \ldots, y_{i_j}\}$.

Gaussian Processes

In this work, we place independent Gaussian process priors on the different objectives. In this section we therefore restrict ourselves to modeling a single objective $f \sim GP(\mu_0, K_0)$, where $\mu_0 : \mathcal{X} \to \mathbb{R}$ is the prior function (assumed to be constant) and $K_0 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the the prior covariance function. We assume that observations of the objectives are subject to iid zero-mean Gaussian noise with variance σ^2. Then after conditioning on $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ observations, the mean and covariance functions conditioned on $\mathcal{D}$ at a set of points $\boldsymbol{x}$ are given by [Rasmussen, 2004]

$$\mu_{\mathcal{D}}(\boldsymbol{x}) = \mu_0(\boldsymbol{x}) + K_0(\boldsymbol{x}, \boldsymbol{x}_{1:n})[K_0^{\sigma}(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})]^{-1}[y_{1:n} - \mu_0(\boldsymbol{x}_{1:n})]$$

$$K_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{x}') = K_0(\boldsymbol{x}, \boldsymbol{x}') - K_0(\boldsymbol{x}, \boldsymbol{x}_{1:n})[K_0^{\sigma}(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})]^{-1} K_0(\boldsymbol{x}_{1:n}, \boldsymbol{x}'),$$

where $\boldsymbol{x}_{1:n} := \{x_1, \ldots, x_n\}$, $K_{\mathcal{D}}^{\sigma}(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})$ denotes $K_{\mathcal{D}}(\boldsymbol{x}_{1:n}, \boldsymbol{x}_{1:n})$+diag$(\sigma^2(x_1), ..., \sigma^2(x_n))$.

In this work, we are often interested in fantasization; i.e. fantasizing about the observations $\boldsymbol{y}$ that we would receive if we were to evaluate $\boldsymbol{x}$. In this case, $\boldsymbol{y}$ is a random vector, which according to our beliefs is $y \sim \mathcal{N}(\mu_{\mathcal{D}}(\boldsymbol{x}), K_{\mathcal{D}}(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2(\boldsymbol{x}))$. Then conditioned on evaluating $\boldsymbol{x}$ and observing $\boldsymbol{y}$, the updated posterior mean function would be

$$\mu_{\mathcal{D}_{\boldsymbol{x}}}(\boldsymbol{x}') = \mu_{\mathcal{D}}(\boldsymbol{x}') + K_{\mathcal{D}}(\boldsymbol{x}', \boldsymbol{x})[K_{\mathcal{D}}^{\sigma}(\boldsymbol{x}, \boldsymbol{x})]^{-1}[\boldsymbol{y} - \mu_{\mathcal{D}}(\boldsymbol{x})].$$

As in previous works, it is convenient to express the updated mean in terms of a standard normal random variable [Wu et al., 2020a, Wu and Frazier, 2016]. We can rewrite $K_{\mathcal{D}}^{\sigma}(\boldsymbol{x}, \boldsymbol{x})$ in terms of its Cholesky factors $K_{\mathcal{D}}^{\sigma}(\boldsymbol{x}, \boldsymbol{x}) = L_{\mathcal{D}}(\boldsymbol{x}) L_{\mathcal{D}}(\boldsymbol{x})^T$. So $[K_{\mathcal{D}}^{\sigma}(\boldsymbol{x}, \boldsymbol{x})]^{-1} = (L_{\mathcal{D}}(\boldsymbol{x})^T)^{-1} L_{\mathcal{D}}(\boldsymbol{x})^{-1}$. Since $[\boldsymbol{y} - \mu_{\mathcal{D}}(\boldsymbol{x})] \sim \mathcal{N}(0, K_{\mathcal{D}}^{\sigma}(\boldsymbol{x}, \boldsymbol{x}))$, $L_{\mathcal{D}}(\boldsymbol{x})^{-1}[\boldsymbol{y} - \mu_{\mathcal{D}}(\boldsymbol{x})]$ is a standard normal random vector. Letting $\hat{\Sigma}_{\mathcal{D}}(\boldsymbol{x}', \boldsymbol{x}) := K_{\mathcal{D}}(\boldsymbol{x}', \boldsymbol{x})(L_{\mathcal{D}}(\boldsymbol{x})^T)^{-1}$, we can express the update posterior mean as

$$\mu_{\mathcal{D}_{\boldsymbol{x}}}(\boldsymbol{x}') = \mu_{\mathcal{D}}(\boldsymbol{x}') + \hat{\Sigma}_{\mathcal{D}}(\boldsymbol{x}', \boldsymbol{x})\epsilon,$$

where ϵ is a standard normal random vector.

5.B.2 Proofs

Without loss of generality, we consider case with a batch size $q = 1$ (i.e. $\mathcal{X}_{\text{cand}} = \{x\}$). Since $\mathcal{X}_{\text{cand}}$ only affects the new data $\mathcal{D}_{\boldsymbol{x}}$ that the model is conditioned on, partial derivatives can be computed for all $q \cdot d$ elements of $\mathcal{X}_{\text{cand}}$ and extending the results that follow is straightforward.[14] Moreover, for brevity we only consider (iid) Monte Carlo sampling in this section. Balandat et al. [2020] also prove basic results for SAA using (randomized) quasi-Monte Carlo (RQMC) sampling; leveraging those results the proofs in this section can be extended to the RQMC setting in a straightforward fashion.

[14] Note that there is a minor complication if $\mathcal{X}_{\text{cand}}$ contains duplicate points as the posterior mean will not be differentiable at such points. However, the set of such points is of measure zero and so does not affect the derivations and the results below.

At a high level, we derive a gradient estimator and prove that it is unbiased (Theorem 5.B.1) by building upon the work of Wu et al. [2020b] and leveraging our proof that value function in HV-KG is Lipschitz continuous (Lemma 5.B.1). Then, we prove our main result (Theorem 5.8.1), which proves three convergence properties of our SAA estimator, building upon work from Balandat et al. [2020] and leveraging Lemma 5.B.2.

Lemma 5.B.1. *For a fixed X, let $A(\boldsymbol{x}, \boldsymbol{\epsilon}) := HV\Big[\boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X)\Big]$, where $\boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X) := \Big[\mu^{(1)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X), \dots, \mu^{(M)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X)\Big]$, $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X) := \mu^{(m)}_{\mathcal{D}}(X) + \hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{x})\epsilon^{(m)}$ for $m = 1, \dots M$, and $\boldsymbol{\epsilon} := [\epsilon^{(1)} ..., \epsilon^{(M)}]$. Then, $A(\boldsymbol{x}, \boldsymbol{\epsilon})$ is Lipschitz continuous with respect to $\boldsymbol{x}$ for any given $\boldsymbol{\epsilon}$.*

Proof. Note that

$$A(\boldsymbol{x}, \boldsymbol{\epsilon}) = \mathrm{HV}\Big[\boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X)\Big] = \sum_{j=1}^{N_p} \sum_{X_j \in \mathbb{X}_j} (-1)^{j+1} \prod_{m=1}^{M} \Big[\min[\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_1}), \dots, \mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_j})] - r^{(m)}\Big]_+,$$

where $\mathbb{X}_j := \{X_j \subseteq X : |X_j| = j\}$ is the set of all subsets of X of size j.

We wish to show that there exists a function $l : \mathbb{R}^M \to \mathbb{R}$ such that $|A(\boldsymbol{x}, \boldsymbol{\epsilon}) - A(\boldsymbol{y}, \boldsymbol{\epsilon})| \leq l(\boldsymbol{\epsilon})||\boldsymbol{x} - \boldsymbol{y}||$.

Let $\tilde{a}_{m,j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) = \Big[\min[\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_1}), ..., \mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_j})] - r^{(m)}\Big]_+$ and let $\tilde{A}_{j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) = \prod_{m=1}^{M} \tilde{a}_{m,j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon})$. Since $A(\boldsymbol{x}, \boldsymbol{\epsilon}) = \sum_{j=1}^{N_p} \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \tilde{A}_{j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon})$, it suffices to show that there exists a function $l : \mathbb{R}^M \to \mathbb{R}$ such that $|\tilde{A}_{j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) - \tilde{A}_{j,X_j}(\boldsymbol{y}, \boldsymbol{\epsilon})| \leq l(\boldsymbol{\epsilon})||\boldsymbol{x} - \boldsymbol{y}||$.

We have that

$$\begin{aligned}
\tilde{a}_{m,j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) &= \Big[\min[\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_1}), ..., \mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_j})] - r^{(m)}\Big]_+ \\
&\leq |r^{(m)}| + \Big|\min[\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_1}), ..., \mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_j})]\Big| \\
&\leq |r^{(m)}| + \sum_{k=1}^{j} \Big|\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k})\Big|.
\end{aligned}$$

Note that for a given $\boldsymbol{\epsilon}$, $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X)$ is continuously differentiable with respect to $\boldsymbol{x}$ for any fixed X and continuously differentiable w.r.t to X for any $\boldsymbol{x}$ because $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X) = \mu^{(m)}_{\mathcal{D}}(X) + \hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{x})\epsilon^{(m)}$ and $\mu^{(m)}_{\mathcal{D}}(X)$ and $\hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{x})$ are continuously

differentiable with respect to $\boldsymbol{x}$ [Wu et al., 2020a].[15] Note that $|\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X)| \leq ||\mu^{(m)}_{\mathcal{D}}(X)|| + ||\hat{\Sigma}^{(m)}_{\mathcal{D}}(X,\boldsymbol{x})|| \cdot |\epsilon^{(m)}|$. Since $\mu^{(m)}_{\mathcal{D}}(X)$ and $\hat{\Sigma}^{(m)}_{\mathcal{D}}(X,\boldsymbol{x})$ are uniformly bounded for each $m = 1,\ldots,M$, there exist $C^{(m)}_1, C^{(m)}_2 \in \mathbb{R}$ such that $|\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X)| \leq C^{(m)}_1 + C^{(m)}_2|\epsilon^{(m)}|$ for each $m = 1,\ldots,M$. Hence, $|\tilde{a}_{m,j,X_j}(\boldsymbol{x},\boldsymbol{\epsilon})| \leq |r^{(m)}| + j(C^{(m)}_1 + C^{(m)}_2|\epsilon^{(m)}|)$.

Omitting the subscripts j, X_j for brevity, and considering $M = 2$ for now, we have that

$$
\begin{aligned}
&|\tilde{A}_{j,X_j}(\boldsymbol{x},\boldsymbol{\epsilon}) - \tilde{A}_{j,X_j}(\boldsymbol{y},\boldsymbol{\epsilon})| && (5.10)\\
&\quad = \Big|\tilde{a}_1(\boldsymbol{x},\boldsymbol{\epsilon})\tilde{a}_2(\boldsymbol{x},\boldsymbol{\epsilon}) - \tilde{a}_1(\boldsymbol{y},\boldsymbol{\epsilon})\tilde{a}_2(\boldsymbol{y},\boldsymbol{\epsilon})\Big| && (5.11)\\
&\quad = \Big|\tilde{a}_1(\boldsymbol{x},\boldsymbol{\epsilon})((\tilde{a}_2(\boldsymbol{x},\boldsymbol{\epsilon}) - \tilde{a}_2(\boldsymbol{y},\boldsymbol{\epsilon})) + \tilde{a}_2(\boldsymbol{y},\boldsymbol{\epsilon})(\tilde{a}_1(\boldsymbol{x},\boldsymbol{\epsilon}) - \tilde{a}_1(\boldsymbol{y},\boldsymbol{\epsilon}))\Big| && (5.12)\\
&\quad \leq \Big|\tilde{a}_1(\boldsymbol{x},\boldsymbol{\epsilon})\Big|\Big|\tilde{a}_2(\boldsymbol{x},\boldsymbol{\epsilon}) - \tilde{a}_2(\boldsymbol{y},\boldsymbol{\epsilon})\Big| + \Big|\tilde{a}_2(\boldsymbol{y},\boldsymbol{\epsilon})\Big|\Big|\tilde{a}_1(\boldsymbol{x},\boldsymbol{\epsilon}) - \tilde{a}_1(\boldsymbol{y},\boldsymbol{\epsilon})\Big|. && (5.13)
\end{aligned}
$$

Note that

$$
\begin{aligned}
&|a_{m,j,X_j}(\boldsymbol{x},\boldsymbol{\epsilon}) - a_{m,j,X_j}(\boldsymbol{y},\boldsymbol{\epsilon})|\\
&\qquad = \Bigg|\Big[\min[\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(\boldsymbol{x}_{i_1}),...,\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(\boldsymbol{x}_{i_j})] - r^{(m)}\Big]_+\\
&- \Big[\min[\mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(\boldsymbol{x}_{i_1}),...,\mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(\boldsymbol{x}_{i_j})] - r^{(m)}\Big]_+\Bigg|.
\end{aligned}
$$

For brevity, we assume without loss of generality that $r = 0$ (otherwise this is just a constant shift in the means μ).

Case 1: If both terms are zero, then $|a_{m,j,X_j}(\boldsymbol{x},\boldsymbol{\epsilon}) - a_{m,j,X_j}(\boldsymbol{y},\boldsymbol{\epsilon})| = 0$.

Case 2: Suppose that one of the terms inside of $[\cdot]_+$ is greater than 0 and one term is less than zero. Without loss of generality suppose that

$$\min[\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_1}),...,\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_j})] \leq 0$$

and

$$\min[\mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_1}),...,\mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_j})] \geq 0.$$

[15] Technically, this is only true if the noise terms $\{\sigma^2(X_i)\}_{i=1}^n$ are strictly positive; otherwise $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X)$ is not differentiable if $K^{\sigma}_0(\boldsymbol{x}_{1:n},\boldsymbol{x}_{1:n})$ is singular. However, even in this case that happens only on a set of measure zero, and thus our arguments remain valid in the almost everywhere sense.

Let $k = \arg\min_{k=1,...,j} \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})$. Recall that $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X) = \mu^{(m)}_{\mathcal{D}}(X) + \hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{x})\epsilon^{(m)}$ and $\hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{x})$ is continuously differentiable with respect to $\boldsymbol{x}$ [Wu et al., 2020a], so they are Lipschitz with respect to $\boldsymbol{x}$. Hence,

$$\begin{aligned}|\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k}) - \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})| &= |\mu^{(m)}_t(X) + \hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{x})\epsilon^{(m)} - \mu^{(m)}_t(X) - \hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{y})\epsilon^{(m)}| \\ &= |\hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{x})\epsilon^{(m)} - \hat{\Sigma}^{(m)}_{\mathcal{D}}(X, \boldsymbol{y})\epsilon^{(m)}| \\ &\leq C^{(m)}_3|\epsilon^{(m)}| \cdot ||\boldsymbol{x} - \boldsymbol{y}||,\end{aligned}$$

where $C^{(m)}_3 \in \mathbb{R}$, for all $m = 1, ..., M$. Note that $\mu_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k}) \leq 0 \leq \mu_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})$. So $|a_{m,j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) - a_{m,j,X_j}(\boldsymbol{y}, \boldsymbol{\epsilon})| = |0 - \mu_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})| \leq |\mu_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k}) - \mu_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})| \leq C^{(m)}_3|\boldsymbol{\epsilon}| \cdot ||\boldsymbol{x} - \boldsymbol{y}||$.

Case 3: Suppose both terms are not zero, i.e., $\min[\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_1}), ..., \mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_j})] \geq 0$ and $\min[\mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_1}), ..., \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_j})] \geq 0$. Let $k = \arg\min_{k=1,...,j} \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})$. Let $q = \arg\min_{k=1,...,j} \mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k})$.

Suppose $k = q$. Then, $|a_{m,j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) - a_{m,j,X_j}(\boldsymbol{y}, \boldsymbol{\epsilon})| = |\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k}) - \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})| \leq C^{(m)}_3|\epsilon^{(m)}| \cdot ||\boldsymbol{x} - \boldsymbol{y}||$.

Suppose $k \neq q$.

Suppose $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_q}) \leq \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})$. Since $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_q}) \leq \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k}) \leq \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_q})$, we have that $|\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_q}) - \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})| \leq |\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k}) - \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})| \leq C^{(m)}_3|\epsilon^{(m)}| \cdot ||\boldsymbol{x} - \boldsymbol{y}||$ because $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(\cdot)$, $\mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(\cdot)$ are Lipschitz w.r.t. $\boldsymbol{x}$, $\boldsymbol{y}$ respectively, as noted above.

Suppose $\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_q}) > \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})$. Similarly, since $\mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k}) < \mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_q}) \leq \mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k})$, we have that $|\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_q}) - \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})| \leq |\mu^{(m)}_{\boldsymbol{x},\boldsymbol{\epsilon}}(X_{i_k}) - \mu^{(m)}_{\boldsymbol{y},\boldsymbol{\epsilon}}(X_{i_k})| \leq C^{(m)}_3|\epsilon^{(m)}| \cdot ||\boldsymbol{x} - \boldsymbol{y}||$.

So, $|a_{m,j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) - a_{m,j,X_j}(\boldsymbol{y}, \boldsymbol{\epsilon})| \leq C^{(m)}_3|\epsilon^{(m)}| \cdot ||\boldsymbol{x} - \boldsymbol{y}||$.

Hence, in all cases, $|a_{m,j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) - a_{m,j,X_j}(\boldsymbol{y}, \boldsymbol{\epsilon})| \leq C^{(m)}_3|\epsilon^{(m)}| \cdot ||\boldsymbol{x} - \boldsymbol{y}||$.

Plugging into (5.13), we have

$$|\tilde{A}_{j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon}) - \tilde{A}_{j,X_j}(\boldsymbol{y}, \boldsymbol{\epsilon})| \leq l(\boldsymbol{\epsilon})||\boldsymbol{x} - \boldsymbol{y}||.$$

with

$$l(\boldsymbol{\epsilon}) = \Big(|r^{(1)}| + j(C_1^{(1)} + C_2^{(1)}|\epsilon^{(1)}|)\Big) \cdot C_3^{(2)}|\epsilon^{(2)}| + (|r^{(2)}| + j\Big(C_1^{(2)} + C_2^{(2)}|\epsilon^{(2)}|)\Big) \cdot C_3^{(1)}|\epsilon^{(1)}|).$$

This result can be generalized for any M by telescoping the expressions in (5.11). Hence $\tilde{A}_{j,X_j}(\boldsymbol{x}, \boldsymbol{\epsilon})$ is $l(\boldsymbol{\epsilon})$-Lipschitz continuous and thus $A(\boldsymbol{x}, \boldsymbol{\epsilon})$ is Lipschitz continuous. □

Theorem 5.B.1. *Let the search space $\mathcal{X}$ be compact, the prior mean function μ_0 be constant, and the prior covariance function K_0 be continuously differentiable. Let $X^* \in \arg\max_{X \subseteq \mathcal{X}} HV\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]$. Then*

$$\nabla_{\boldsymbol{x}} \mathbb{E}_{\mathcal{D}}\left[\max_{X \subseteq \mathcal{X}} HV\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]\right] = \mathbb{E}_{\mathcal{D}}\left[\nabla_{\boldsymbol{x}} HV\Big[\boldsymbol{\mu}(X^* \mid \mathcal{D}_{\boldsymbol{x}})\Big]\right].$$

Proof. The proof follows that of [Wu et al., 2020a, Theorem 1]. We wish to show that

$$\nabla_{\boldsymbol{x}} \mathbb{E}_{\mathcal{D}}\left[\max_{X \subseteq \mathcal{X}} \mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]\right] = \mathbb{E}_{\mathcal{D}}\left[\nabla_{\boldsymbol{x}} \max_{X \subseteq \mathcal{X}} \mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]\right] \tag{5.14}$$

$$= \mathbb{E}_{\mathcal{D}}\left[\nabla_{\boldsymbol{x}} \mathrm{HV}\Big[\boldsymbol{\mu}(X^* \mid \mathcal{D}_{\boldsymbol{x}})\Big]\right]. \tag{5.15}$$

To justify (5.15), we begin by expressing the posterior mean

$$\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}}) = \Big[\mu^{(1)}(X \mid \mathcal{D}_{\boldsymbol{x}}), \ldots, \mu^{(M)}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]$$

for each outcome in terms of a standard normal random vector: $\mu^{(m)}(X) = \mu^{(m)}(X \mid \mathcal{D}) + \hat{\Sigma}_{\mathcal{D}}^{(m)}(X, \boldsymbol{x} \mid \mathcal{D})\epsilon^{(m)}$ for $m = 1, ..., M$. Note that for a fixed $\boldsymbol{x}$, $\mu^{(m)}(X \mid \mathcal{D})$ and $\Sigma^{(m)}(X, \boldsymbol{x} \mid \mathcal{D})$ are continuously differentiable in X a.e.,[16] and for a fixed X, $\Sigma^{(m)}(X, \boldsymbol{x} \mid \mathcal{D})$ is continuously differentiable in $\boldsymbol{x}$ [Wu et al., 2020a, Lemma 1].[0] Hence, $\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})$ is continuously differentiable w.r.t to $\boldsymbol{x}$ for fixed X and continuously differentiable w.r.t to X for fixed $\boldsymbol{x}$.[0]

From Equation (5.9), it is easily to see that $\mathrm{HV}(Y, r)$ is continuous over $Y \in \mathbb{R}^M$. Moreover, the partial derivatives of the input Y exist almost everywhere, since HV

[16] If there are repeated points in X and noise variance is not positive at all points X, then the gradient does not exist.

is an incarnation of hypervolume improvement with no incumbent Pareto frontier and hypervolume improvement is differentiable almost everywhere w.r.t Y [Daulton et al., 2020].[17] Hence, $\mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]$ is differentiable w.r.t $\boldsymbol{x}$ almost everywhere for a fixed X and $\boldsymbol{\epsilon}$ and is differentiable w.r.t X almost everywhere for a fixed $\boldsymbol{x}$ and $\boldsymbol{\epsilon}$.

To employ the envelope theorem [Milgrom and Segal, 2002], we need to show that the following conditions of Milgrom and Segal [2002, Theorem 2] hold:

1. $\mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]$ is absolutely continuous w.r.t. $\boldsymbol{x}$ for a fixed $\boldsymbol{\epsilon}$ and a fixed X.
2. There exists an integrable function $b : \mathcal{X} \to \mathbb{R}$ such that $||\nabla_{\boldsymbol{x}} \mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]|| \leq b(\boldsymbol{x})$ for almost all $\boldsymbol{x} \in \mathcal{X}$ and for all X.

From Lemma 5.B.1, $\mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]$ is Lipschitz continuous in $\boldsymbol{x}$ for a fixed $\boldsymbol{\epsilon}, X$, so it is absolutely continuous. Furthermore, since it is Lipschitz continuous, its gradient is bounded almost everywhere. Hence, we have

$$\nabla_{\boldsymbol{x}} \max_{X \subseteq \mathcal{X}} \mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big] = \nabla_{\boldsymbol{x}} \mathrm{HV}\Big[\max_{X \subseteq \mathcal{X}} \boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big] = \nabla_{\boldsymbol{x}} \mathrm{HV}\Big[\boldsymbol{\mu}(X^* \mid \mathcal{D}_{\boldsymbol{x}})\Big],$$

showing equality between (5.14) and (5.15). To show (5.14), we note that since $\mathcal{X}$ is compact, $\mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]$ is bounded, which satisfies the conditions of Bartle [1995, Corollary 5.8]. Given the result in Bartle [1995, Corollary 5.8] and noting again that the partial derivatives of $\mathrm{HV}\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}})\Big]$ are bounded almost everywhere, we can interchange expectation and gradient [Bartle, 1995, Corollary 5.9], which justifies (5.14) and completes the proof. □

Corollary 5.B.1. *Let* $\mu_i^{(m)}(X) := \mu^{(m)}(X \mid \mathcal{D}) + \hat{\Sigma}^{(m)}(X, \boldsymbol{x} \mid \mathcal{D})\epsilon_i^{(m)}$ *for* $m = 1, \ldots, M$, $\boldsymbol{\mu}_i(X) := \Big[\mu_i^{(1)}(X), \ldots, \mu_i^{(M)}(X)\Big]$, *and* $\epsilon_i \sim \mathcal{N}(0, I_M)$ *iid. Let* $X_i^* \in \arg\max_{X \subseteq \mathcal{X}} HV\Big[\boldsymbol{\mu}(X \mid \mathcal{D}_{\boldsymbol{x}}^i)\Big]$,

[17] The partial derivative of HV with respect to $Y_{i,j}$, the element in the i^{th} row and j^{th} column of Y, is not defined when there exists another $k! = i$ such that $Y_{i,j} = Y_{k,j}$ or when $Y_{i,j} = r_j$, where r_j is the reference point value for objective j. The set of points defined by the union of these settings has zero measure under any GP posterior [Daulton et al., 2020]. Furthermore, the gradient is only computed at the optimal $\mathcal{X}'^*$, and typically, $N_p \leq |\mathcal{X}^*|$, so columns of $\boldsymbol{\mu}_{t+1}(\mathcal{X}'^*)$ will contain unique values if $\boldsymbol{\mu}_{t+1}$ is representative of the underlying objectives.

where $\mathcal{D}_{\boldsymbol{x}}^i = \mathcal{D} \cup \{(\boldsymbol{x}, \boldsymbol{y}^i)\}$ *with* $\boldsymbol{y}^i \sim p(\boldsymbol{y} \,|\, \mathcal{D}, \boldsymbol{x})$*. Then an unbiased estimator of the gradient* $\nabla_{\boldsymbol{x}} \alpha_{HV\text{-}KG}(\boldsymbol{x})$ *is given by the average of the sample-level gradients*

$$\nabla_{\boldsymbol{x}} \alpha_{HV\text{-}KG}(\boldsymbol{x}) \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{x}} HV\Big[\boldsymbol{\mu}(X_i^* \,|\, \mathcal{D}_{\boldsymbol{x}}^i)\Big].$$

The result follow directly from Theorem 5.B.1 and approximating the expectation via Monte Carlo (using independence of the ϵ_i). Computing the gradient estimator in Corollary 5.B.1 requires solving the inner maximization problem to obtain X_i^* and computing the sample-level gradient for each of the N samples.

Lemma 5.B.2. *Suppose that* $\mathcal{X}$ *is compact and that* $\boldsymbol{f}(\boldsymbol{x}) \sim GP(\boldsymbol{0}, K_0(\boldsymbol{x}, \boldsymbol{x}))$ *is a zero-mean multi-output Gaussian Process prior with* M *outputs. Suppose that* $||\boldsymbol{f}(\boldsymbol{x})|| < \infty$ *almost surely for all* $\boldsymbol{x} \in \mathcal{X}$*. Let* $X \subseteq \mathcal{X}$ *such that* $|X| \leq N_p$*, and let* $r \in \mathbb{R}^M$*. Then, the moment generating function*

$$\mathbb{E}\bigg[\exp\Big(t \cdot \sup_{X \subseteq \mathcal{X}} HV\big[\boldsymbol{f}(X)\big]\Big)\bigg]$$

of $\sup_{X \subseteq \mathcal{X}} HV[\boldsymbol{f}(X)]$*, where* $t \in \mathbb{R}$*, is finite for all* t*.*

Proof. Let use denote the components of $\boldsymbol{f}(\boldsymbol{x})$ by $f^{(1)}(\boldsymbol{x}), ..., f^{(M)}(\boldsymbol{x})$. Since $||\boldsymbol{f}(\boldsymbol{x})|| < \infty$ for all $\boldsymbol{x} \in \mathcal{X}$ a.s., we have that $|f^{(i)}(\boldsymbol{x})| < \infty$ for all $\boldsymbol{x} \in \mathcal{X}$ and $m = 1, \ldots, M$ a.s.. Therefore, $\mathbb{E}\big[\sup_{\boldsymbol{x} \in \mathcal{X}} f^{(m)}(\boldsymbol{x})] < \infty$ for $m = 1, \ldots, M$ [Adler, 1990, Theorem 2.1]. Let $\boldsymbol{f}^*$ denote the component-wise supremum of $\boldsymbol{f}$: i.e. $\boldsymbol{f}^* = [\sup_{\boldsymbol{x} \in \mathcal{X}} f^{(1)}(\boldsymbol{x}), ..., \sup_{\boldsymbol{x} \in \mathcal{X}} f^{(M)}\boldsymbol{x})]$. By definition $\boldsymbol{f}^* \succeq \boldsymbol{f}(\boldsymbol{x})$ for all $\boldsymbol{x} \in X$. Hence, $\mathrm{HV}[\{\boldsymbol{f}^*\}] \geq \sup_{X \subseteq \mathcal{X}} \mathrm{HV}[\boldsymbol{f}(X)]$. Since HV is non-negative, it is sufficient to consider $t \geq 0$. From Equation (5.9), we have that $\mathrm{HV}[\{\boldsymbol{f}^*\}] = \prod_{m=1}^{M} \max(\sup_{\boldsymbol{x} \in \mathcal{X}} f^{(m)}(\boldsymbol{x}) - r^{(m)}, 0)$. Without loss of generality, we may assume $r = 0$ (otherwise this corresponds to a simple shift of $\boldsymbol{f}$). Then,

$$\begin{aligned}
\mathrm{HV}(\boldsymbol{f}^*) &= \prod_{m=1}^{M} \max\Big(\sup_{\boldsymbol{x} \in \mathcal{X}} f^{(m)}(\boldsymbol{x}), 0\Big) \\
&\leq \bigg(\max_{m=1...,M}\Big[\max\Big(\sup_{\boldsymbol{x} \in \mathcal{X}} f^{(m)}(\boldsymbol{x}), 0\Big)\Big]\bigg)^M \\
&\leq \max_{m=1...,M}\Big|\sup_{\boldsymbol{x} \in \mathcal{X}} f^{(m)}(\boldsymbol{x})\Big|^M \\
&\leq \max_{m=1...,M}\Big(\sup_{\boldsymbol{x} \in \mathcal{X}} |f^{(m)}(\boldsymbol{x})|\Big)^M.
\end{aligned}$$

Hence

$$\mathbb{E}\Big[\exp\Big(t\cdot \mathrm{HV}(\boldsymbol{f}^*)\Big)\Big] \leq \mathbb{E}\Big[\exp\Big(t\cdot \max_{m=1\ldots,M}\Big(\sup_{\boldsymbol{x}\in\mathcal{X}}|f^{(m)}(\boldsymbol{x})|\Big)^M\Big)\Big] \tag{5.16}$$

$$= \mathbb{E}\Big[\max_{m=1\ldots,M}\exp\Big(t\cdot\Big(\sup_{\boldsymbol{x}\in\mathcal{X}}|f^{(m)}(\boldsymbol{x})|\Big)^M\Big)\Big] \tag{5.17}$$

$$\leq \sum_{m=1}^{M}\mathbb{E}\Big[\exp\Big(t\cdot\Big(\sup_{\boldsymbol{x}\in\mathcal{X}}|f^{(m)}(\boldsymbol{x})|\Big)^M\Big)\Big], \tag{5.18}$$

where the final inequality comes from noting that all terms in the max are positive. Since all moments of $\sup_{\boldsymbol{x}\in\mathcal{X}}|f^{(m)}(\boldsymbol{x})|$ are finite [Balandat et al., 2020, Lemma 4], $\mathbb{E}\Big[\sup_{\boldsymbol{x}\in\mathcal{X}}|f^{(m)}(\boldsymbol{x})|^M\Big] \leq \infty$ for all $m = 1, ..., M$. From here, our proof follows that of Balandat et al. [2020, Lemma 4]. Consider the m^{th} term in (5.18) and let $Z^{(m)} := \Big(\sup_{\boldsymbol{x}\in\mathcal{X}}|f^{(m)}(\boldsymbol{x})|\Big)^M$:

$$\begin{aligned}
\mathbb{E}[\exp(t\cdot Z^{(m)})] &= \int_0^\infty p\Big(\exp(t\cdot Z^{(m)}) > u\Big)du \\
&\leq 1 + \int_1^\infty p\Big(\exp(t\cdot Z^{(m)}) > u\Big)du \\
&= 1 + \int_1^\infty p\Big(Z^{(m)} > \frac{\log u}{t}\Big)du \\
&= 1 + \int_1^\infty p\Big(Z^{(m)} - \mathbb{E}[Z^{(m)}] > \frac{\log u}{t} - \mathbb{E}[Z^{(m)}]\Big)du
\end{aligned}$$

Using a change of variables where $v = \frac{\log u}{t} - \mathbb{E}[Z^{(m)}]$, we have that $dv = \frac{du}{ut}$ and $ut = te^{tv}e^{t\mathbb{E}[Z^{(m)}]}$. Hence via substitution,

$$\begin{aligned}
\mathbb{E}[\exp(t\cdot Z^{(m)})] &\leq 1 + \int_1^\infty p\Big(Z^{(m)} - \mathbb{E}[Z^{(m)}] > \frac{\log u}{t} - \mathbb{E}[Z^{(m)}]\Big)du \\
&= 1 + te^{\mathbb{E}[Z^{(m)}]}\int_{-\mathbb{E}[Z^{(m)}]}^\infty p\Big(Z^{(m)} - \mathbb{E}[Z^{(m)}] > v\Big)e^{tv}dv \\
&= 1 + te^{\mathbb{E}[Z^{(m)}]}\Bigg[\int_{\min(-\mathbb{E}[Z^{(m)}],0)}^0 p\Big(Z^{(m)} - \mathbb{E}[Z^{(m)}] > v\Big)e^{tv}dv \\
&\qquad + \int_0^\infty p\Big(Z^{(m)} - \mathbb{E}[Z^{(m)}] > v\Big)e^{tv}dv\Bigg] \\
&\leq 1 + te^{\mathbb{E}[Z^{(m)}]}\Bigg[|\mathbb{E}[Z^{(m)}]| + \int_0^\infty p\Big(Z^{(m)} > v + \mathbb{E}[Z^{(m)}]\Big)e^{tv}dv\Bigg]
\end{aligned}$$

Note that since $Z^{(m)} = \Big(\sup_{\boldsymbol{x}\in\mathcal{X}}|f^{(m)}(\boldsymbol{x})|\Big)^M$,

$$p\Big(Z^{(m)} > v + \mathbb{E}[Z^{(m)}]\Big) = p\Big(\sup_{\boldsymbol{x}\in\mathcal{X}}|f^{(m)}(\boldsymbol{x})| > (v + \mathbb{E}[Z^{(m)}])^{\frac{1}{M}}\Big).$$

Let $\sigma^2_{\mathcal{X}} = \sup_{\boldsymbol{x}\in\mathcal{X}} \left[(f^{(m)}(\boldsymbol{x}))^2\right]$. Then, the tail probability $p\Big(\sup_{\boldsymbol{x}\in\mathcal{X}} f^{(m)}(\boldsymbol{x}) > \alpha\Big)$ can be bounded as

$$p\Big(\sup_{\boldsymbol{x}\in\mathcal{X}} f^{(m)}(\boldsymbol{x}) > \alpha\Big) \leq e^{-\alpha^2/(2\sigma^2_{\mathcal{X}})}$$

by Borell's inequality [Adler, 1990, Section 2.1]. Hence,

$$p\Big(\sup_{\boldsymbol{x}\in\mathcal{X}} |f^{(m)}(\boldsymbol{x})| > \alpha\Big) \leq 2e^{-\alpha^2/(2\sigma^2_{\mathcal{X}})}.$$

Letting $\alpha = v + \mathbb{E}[Z^{(m)}]$, we have that

$$p\Big(Z^{(m)} > v + \mathbb{E}[Z^{(m)}]\Big) \leq 2e^{-(v+\mathbb{E}[Z^{(m)}])^2/(2\sigma^2_{\mathcal{X}})}.$$

Hence we obtain

$$\begin{aligned}
\mathbb{E}[\exp(t\cdot Z^{(m)})] &\leq 1 + te^{\mathbb{E}[Z^{(m)}]}\left[|\mathbb{E}[Z^{(m)}]| + \int_0^\infty p\Big(Z^{(m)} > v + \mathbb{E}[Z^{(m)}]\Big)e^{tv}dv\right] \\
&\leq 1 + te^{\mathbb{E}[Z^{(m)}]}|\mathbb{E}[Z^{(m)}]| + te^{\mathbb{E}[Z^{(m)}]}\int_0^\infty 2e^{tv-(v+\mathbb{E}[Z^{(m)}])^2/(2\sigma^2_{\mathcal{X}})}dv \\
&< \infty.
\end{aligned}$$

So,

$$\mathbb{E}\left[\exp\left(t\cdot\sup_{X\subseteq\mathcal{X}} \mathrm{HV}\Big[\boldsymbol{f}(X)\Big]\right)\right] < \infty.$$

□

Theorem 5.8.1. *Suppose that $\mathcal{X}$ is compact and that $\boldsymbol{f} \sim GP(\mu_0(\cdot), K_0(\cdot,\cdot))$ is a sample from a multi-output Gaussian process prior with continuously differentiable mean $\mu_0(\cdot)$ and covariance $K_0(\cdot,\cdot)$ functions. Let $\{\epsilon_i\}_{i=1}^N$ be i.i.d. base samples from $\mathcal{N}(0, I_M)$ and let $\boldsymbol{x}^*_N \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \hat{\alpha}^N_{\text{HV-KG}}(\boldsymbol{x})$, then*

(i) $\hat{\alpha}_{\text{HV-KG}}(\boldsymbol{x}^*_N) \to \alpha^*_{\text{HV-KG}}$ *a.s.*

(ii) $\inf_{\boldsymbol{x}^*\in\mathcal{X}^*_{\text{HV-KG}}} ||\boldsymbol{x}^*_N - \boldsymbol{x}^*|| \to 0$ *a.s.*

(iii) $\forall\ \delta > 0, \exists\ K < \infty, \alpha > 0$ *such that*

$$p\Big(\inf_{\boldsymbol{x}^*\in\mathcal{X}^*_{\text{HV-KG}}} ||\boldsymbol{x}^*_N - \boldsymbol{x}^*|| \geq \delta\Big) \leq Ke^{-\alpha N}.$$

Proof. Let us express the integrand in $\hat{\alpha}^N_{\text{HV-KG}}(\boldsymbol{x})$ as $G(\boldsymbol{x}, \boldsymbol{\epsilon}) = \max_{X \subseteq \mathcal{X}} \text{HV}(\boldsymbol{\mu}_i(X))$, where $\boldsymbol{\mu}_i(\cdot)$ is defined in Corollary 5.B.1. As in Balandat et al. [2020], Daulton et al. [2020], we leverage Homem-de-Mello [2008, Proposition 2.2] to obtain our (i) and (ii). Homem-de-Mello [2008, Proposition 2.2] requires that two conditions be met [Homem-de-Mello, 2008, Assumptions A1, A2]:

(A1) $\forall \boldsymbol{x} \in \mathcal{X},\ \hat{\alpha}^N_{\text{HV-KG}}(\boldsymbol{x}) \to \alpha_{\text{HV-KG}}(\boldsymbol{x})$ a.s.

(A2) there exists an integrable function $L(\boldsymbol{\epsilon}) : \mathbb{R}^M \to \mathbb{R}$ such that for almost every $\boldsymbol{\epsilon}$ and $\forall\ \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$,

$$|G(\boldsymbol{x}, \boldsymbol{\epsilon}) - G(\boldsymbol{y}, \boldsymbol{\epsilon})| \leq L(\boldsymbol{\epsilon})||\boldsymbol{x} - \boldsymbol{y}||.$$

Note that for any $\boldsymbol{\epsilon}$, the restriction from $\boldsymbol{x} \to G(\boldsymbol{x}, \boldsymbol{\epsilon})$ to the k^{th} coordinate, where $\boldsymbol{x} = (x_1, \ldots, x_d)$ and $k \in \{1, ..., d\}$, is Lipschitz continuous by Theorem 5.B.1. Therefore, the partial derivative $\frac{\partial G(\boldsymbol{x}, \boldsymbol{\epsilon})}{\partial x_k}$ exists and is bounded almost everywhere. That is, there exists $c_k \in \mathbb{R}^M$ such that $||c_k|| < \infty$ and $|\frac{\partial G(\boldsymbol{x}, \boldsymbol{\epsilon})}{\partial x_k}| \leq c_k^T|\boldsymbol{\epsilon}|$, where $|\cdot|$ denotes the component-wise absolute value.

Consider the difference $|G(\boldsymbol{x}, \boldsymbol{\epsilon}) - G(\boldsymbol{y}, \boldsymbol{\epsilon})|$. We can bound this difference by summing the component-wise differences and leveraging the bounded partial derivatives to obtain

$$|G(\boldsymbol{x}, \boldsymbol{\epsilon}) - G(\boldsymbol{y}, \boldsymbol{\epsilon})| \leq \sum_{k=1}^{d} c_k^T|\boldsymbol{\epsilon}| \cdot |x_k - y_k| \leq \max_{k \in \{1,\ldots,d\}} c_k^T|\boldsymbol{\epsilon}| \cdot ||x - y||_1. \tag{5.19}$$

Let $L_1(\boldsymbol{\epsilon}) = \max_{k \in \{1,\ldots,d\}} c_k^T|\boldsymbol{\epsilon}|$. We need only to verify that $L_1(\boldsymbol{\epsilon})$ is integrable. Since $\boldsymbol{\epsilon}$ is a vector of standard Normal random variables,

$$\mathbb{E}\Big[|L_1(\boldsymbol{\epsilon})|\Big] \leq \max_{k \in \{1,\ldots,d\}} \sum_{m=1}^{M} c_k^{(m)} \mathbb{E}\Big[|\epsilon^{(m)}|\Big] = \sqrt{\frac{2}{\pi}} \max_{k \in \{1,\ldots,d\}} ||c_k||_1.$$

So $L_1(\boldsymbol{\epsilon})$ is integrable, and assumption (A2) holds.

Note that $G(\boldsymbol{x}, \boldsymbol{\epsilon})$ is the maximum hypervolume where the objectives are GPs. From Lemma 5.B.2, the moment generating function $\mathbb{E}[e^{tG(\boldsymbol{x}, \boldsymbol{\epsilon})}]$ is finite for all t. Noting that $G(\boldsymbol{x}, \boldsymbol{\epsilon})$ is positive for all $\boldsymbol{x}, \boldsymbol{\epsilon}$, we have that $\mathbb{E}[e^{t|G(\boldsymbol{x}, \boldsymbol{\epsilon})|}]$ is also finite for all t. Hence, all of their absolute moments [Meyer, 2012, Exercise 9.15] and $\mathbb{E}[|G(\boldsymbol{x}, \boldsymbol{\epsilon})|]$ are finite for all $\boldsymbol{x}$. Thus, by the strong law of large numbers

$\hat{\alpha}^N_{\text{HV-KG}}(\boldsymbol{x}) \to \alpha_{\text{HV-KG}}(\boldsymbol{x})$ a.s. where $\{\boldsymbol{\epsilon}_i\}_{i=1}^N$ are i.i.d. Therefore assumption (A1) holds.

To obtain (iii), we additionally need to show that there exists an integrable function $L_2(\boldsymbol{\epsilon}) : \mathbb{R}^M \to \mathbb{R}$ such that $G(\boldsymbol{x}, \boldsymbol{\epsilon})$ is $L_2(\boldsymbol{\epsilon})$-Lipschitz and the moment generating function $\mathbb{E}[e^{tL_2(\boldsymbol{\epsilon})}]$ of $L_2(\boldsymbol{\epsilon})$ is finite in an open neighborhood of $t = 0$ (originally from Homem-de-Mello [2008] and written concisely in Balandat et al. [2020, Proposition 2]). Let us define

$$L_2(\boldsymbol{\epsilon}) := M||\boldsymbol{\epsilon}||_\infty \cdot ||c_k||_\infty \geq \max_{k\in\{1,\dots,d\}} c_k^T|\boldsymbol{\epsilon}|.$$

From (5.19) it follows that $G(\boldsymbol{x}, \boldsymbol{\epsilon})$ is $L_2(\boldsymbol{\epsilon})$-Lipschitz in $\boldsymbol{x}$. Furthermore, $||\boldsymbol{\epsilon}||_\infty \leq ||\boldsymbol{\epsilon}||_1$. So, $L_2(\boldsymbol{\epsilon}) \leq C_1||\boldsymbol{\epsilon}||_1$, where $C_1 := M \cdot ||\boldsymbol{c}_k||_\infty < \infty$. Moreover,

$$\mathbb{E}\left[e^{tL_2(\boldsymbol{\epsilon})}\right] \leq \mathbb{E}\left[e^{tC_1||\boldsymbol{\epsilon}||_1}\right] = \mathbb{E}\left[e^{tC_1\sum_{m=1}^M |\epsilon^{(m)}|}\right] = \mathbb{E}\left[\prod_{m=1}^M e^{tC_1|\epsilon^{(m)}|}\right] = \prod_{m=1}^M \mathbb{E}\left[e^{tC_1|\epsilon^{(m)}|}\right],$$

where we arrive at the last equality since $\epsilon^{(1)}, ..., \epsilon^{(M)}$ are independent. Let $M(t) = \prod_{m=1}^M \mathbb{E}\left[e^{tC_1|\epsilon^{(m)}|}\right]$. Note that $M(t)$ is simply the moment generating function of a folded Normal variable with scale parameter C_1^2, which is finite for all t. Hence $\mathbb{E}[e^{tL_2(\boldsymbol{\epsilon})}] < \infty$ for all t, which completes the proof. □

5.C Alternative Knowledge Gradient Acquisition Functions

HV-KG is strongly motivated by the Bayesian decision theoretic best point selection described in Section 5.5. That is, given a model of the objectives a decision maker will typically wish to infer the Pareto set of optimal designs and select one design from the Pareto set based on their preferences and estimates of the objectives for each design. In many MOBO works that consider inference regret [Hernandez-Lobato et al., 2016, Suzuki et al., 2020, Tu et al., 2022], it is common practice to determine the Pareto set over the search space under the posterior mean. Hence, HV-KG is constructed to be the one-step Bayes optimal acquisition function maximizing the hypervolume of the Pareto set under the posterior mean $\text{HV}(\boldsymbol{\mu}(X))$.

An alternative formulation would be to consider hypervolume as a utility function and seek to maximize the expected utility $\mathbb{E}[\text{HV}(\boldsymbol{f}(X))]$. Although many BO acquisition functions including the the single objective knowledge gradient are formulated as expected utilities, expected hypervolume would be difficult to leverage in a Bayesian decision theoretic framework because it quanitifies the expected utility of a set of points rather than an individual point. In the single objective setting with a utility function $g : \mathbb{R} \to \mathbb{R}$ the point[18] x^* that maximizes the expected utility is given by $x^* = \arg\max_{x \in \mathcal{X}} \mathbb{E}[g(f(x))]$. Hence, it is simple to determine the best point with maximum expected utility in this framework. In the multi-objective setting, the hypervolume indicator is a set function and quantifies the utility of a set of points. Although one could identify the optimal set of points[19] $X^* = \arg\max_{X \in \mathcal{X}} \mathbb{E}[\text{HV}(\boldsymbol{f}(X))]$, selecting a single point from X^* to implement according to one's preferences would be challenging. Although the set X^* would be optimal with respect to the expected hypervolume utility, using the posterior mean to estimate the objectives for each point in X^* may yield confusing results. Namely, the points in X^* would not necessarily in the Pareto optimal under the posterior mean. Hence, the expected utility would be misaligned given the method for selecting the best point. In contrast, maximizing the hypervolume of the posterior mean would directly align with the best point selection method.

Nevertheless, we define and evaluate a KG acquisition function that arises when treating HV as an expected utility.

$$\alpha_{\text{E-HV-KG}}(\boldsymbol{x}) = \mathbb{E}_{\mathcal{D}}\left[\max_{X \subseteq \mathcal{X}} \mathbb{E}\Big[\text{HV}(\boldsymbol{f}(X)) \mid \mathcal{D}_{\boldsymbol{x}}\Big] - \phi_*\right], \tag{5.20}$$

where $\phi^* := \max_{X \subseteq \mathcal{X}} \mathbb{E}\Big[\text{HV}(\boldsymbol{f}(X)) \,|\, \mathcal{D}\Big]$. This acquisition function has the desirable property of non-negativity.

Theorem 5.C.1. *$\alpha_{\text{E-HV-KG}}(\boldsymbol{x})$ is non-negative for all $\boldsymbol{x}$ in $\mathcal{X}^q$*

[18] Technically there could be set of maximizers, but here we consider only one for simplicity.

[19] Rather, one could identify an approximation of the optimal set of designs, as discussed in Section 5.5.

Proof. We have that

$$\begin{aligned}\alpha_{\text{E-HV-KG}}(\boldsymbol{x}) &= \mathbb{E}_{\mathcal{D}}\left[\max_{X\subseteq\mathcal{X}} \mathbb{E}\Big[\text{HV}(\boldsymbol{f}(X)) \mid \mathcal{D}_{\boldsymbol{x}}\Big] - \phi_*\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\max_{X\subseteq\mathcal{X}} \mathbb{E}\Big[\text{HV}(\boldsymbol{f}(X)) \mid \mathcal{D}_{\boldsymbol{x}}\Big]\right] - \phi_*.\end{aligned}$$

The proof is straightforward and follows from the fact that the max function is convex. From Jensen's inequality, we have that

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[\max_{X\subseteq\mathcal{X}} \mathbb{E}\Big[\text{HV}(\boldsymbol{f}(X)) \mid \mathcal{D}_{\boldsymbol{x}}\Big]\right] &\geq \max_{X\subseteq\mathcal{X}} \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}\Big[\text{HV}(\boldsymbol{f}(X)) \mid \mathcal{D}_{\boldsymbol{x}}\Big]\right] \\ &= \max_{X\subseteq\mathcal{X}} \mathbb{E}\Big[\text{HV}(\boldsymbol{f}(X)) \mid \mathcal{D}\Big]] \\ &= \phi_*\end{aligned}$$

Hence, $\alpha_{\text{E-HV-KG}}(\boldsymbol{x}) \geq 0$. □

We leave the analysis of the non-negativity of HV-KG with a multi-output Gaussian process prior to future work. We note that hypervolume is not convex, and that for non-Gaussian priors, simple examples show that it can be negative.

5.C.1 Empirical Evaluation

Computing the expected utility requires Monte Carlo integration, and we evaluate the performance below with 16 samples. The decoupled and multi-fidelity variants are straightforward extensions of $\alpha_{\text{E-HV-KG}}$ using the same conditioning on partial information as with HV-KG.

In Figure 5.C.4, we evaluate $\alpha_{\text{E-HV-KG}}$ on single fidelity benchmarks with coupled evaluations and find that HV-KG typically performs at least as well as $\alpha_{\text{E-HV-KG}}$, but is much faster to optimize. $\alpha_{\text{E-HV-KG}}$ is much more expensive to compute due to the nested Monte Carlo integration and is slow even on a GPU as shown in Table 5.D.6.

In Figures 5.C.1 and 5.C.2, we evaluate a decoupled variant of $\alpha_{\text{E-HV-KG}}$ and similar results with respect to optimization performance, wall times, as shown in Tables 5.D.4 and 5.D.5. We where unable to run $\alpha_{\text{E-HV-KG}}$ on the NAS problem with non-competitive decoupling due to memory issues on a CPU and excessive runtime on a CPU. In Figure 5.C.3, we evaluate a MF variant of $\alpha_{\text{E-HV-KG}}$ and

find that it works quite well, but is quite slow and we were unable to run it on the plasma laser acceleration problem and the ranking problem due to memory issues on a GPU and excessive wall time on a CPU. Wall times are reported in Table 5.D.3.

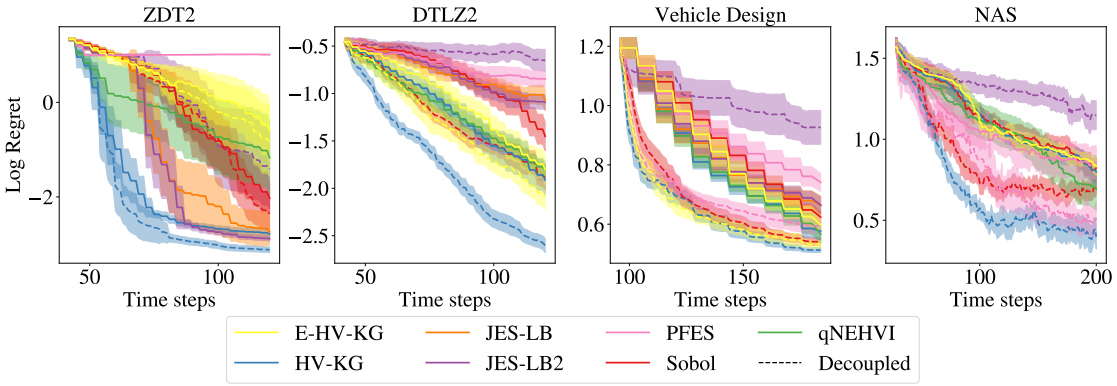

Figure 5.C.1: NCD benchmarks with $\alpha_{\text{E-HV-KG}}$.

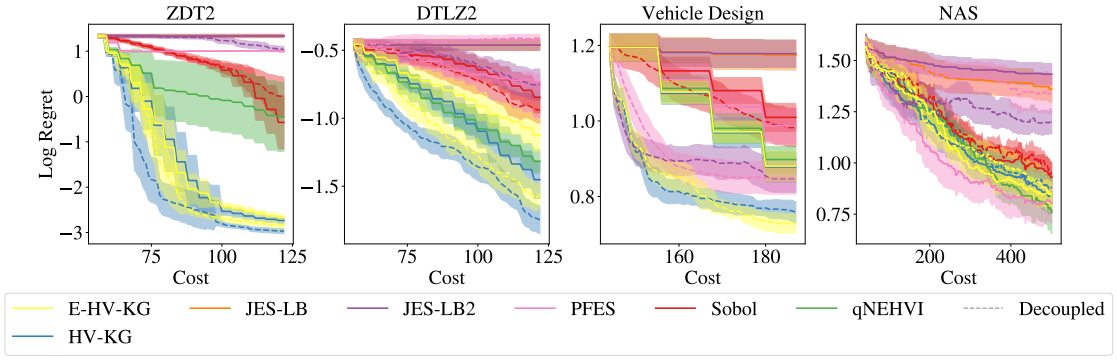

Figure 5.C.2: CD benchmarks with $\alpha_{\text{E-HV-KG}}$.

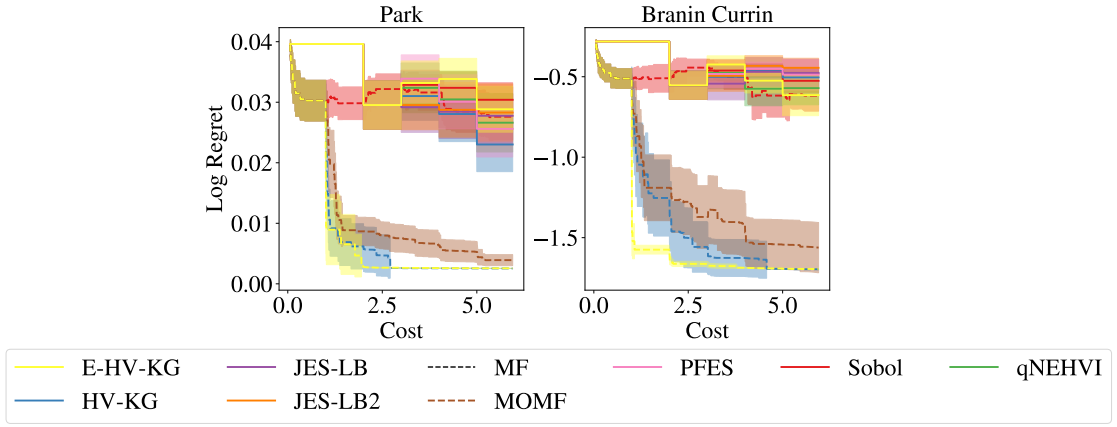

Figure 5.C.3: MF benchmarks with $\alpha_{\text{E-HV-KG}}$.

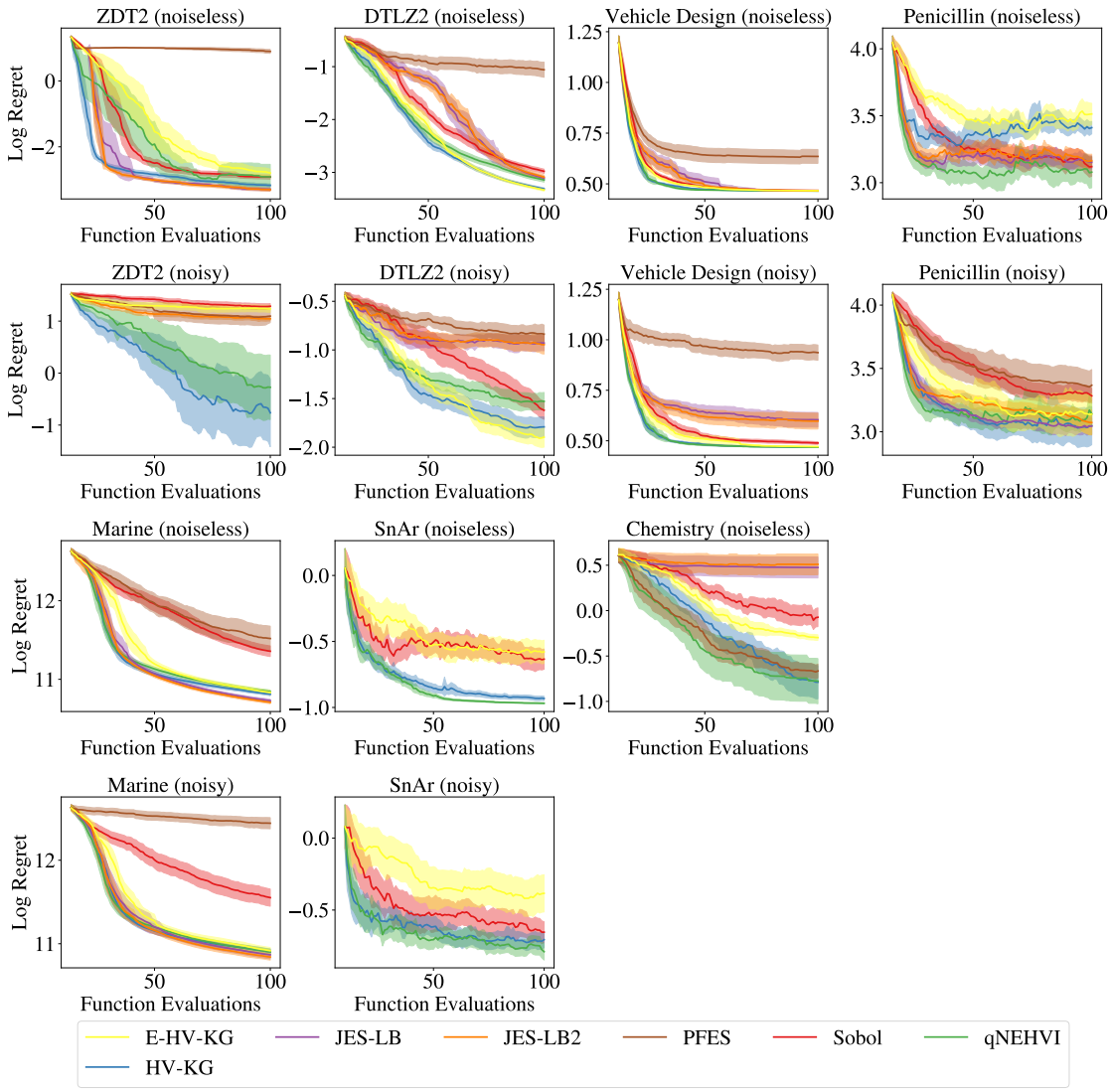

Figure 5.C.4: Single fidelity, coupled evaluation benchmarks with $\alpha_{\text{E-HV-KG}}$.

5.D Additional Experiments

5.D.1 MOBO Problems with Complete Information

Sequential MOBO with Complete Information

We evaluate optimization performance in the standard sequential (i.e. $q = 1$), complete information multi-objective setting (Figure 5.D.1). We find that HV-KG is a top performer on most problems. HV-KG is outperformed by qNEHVI for noiseless Penicilin, and performance is otherwise slightly better than, or not statistically significant from qNEHVI.

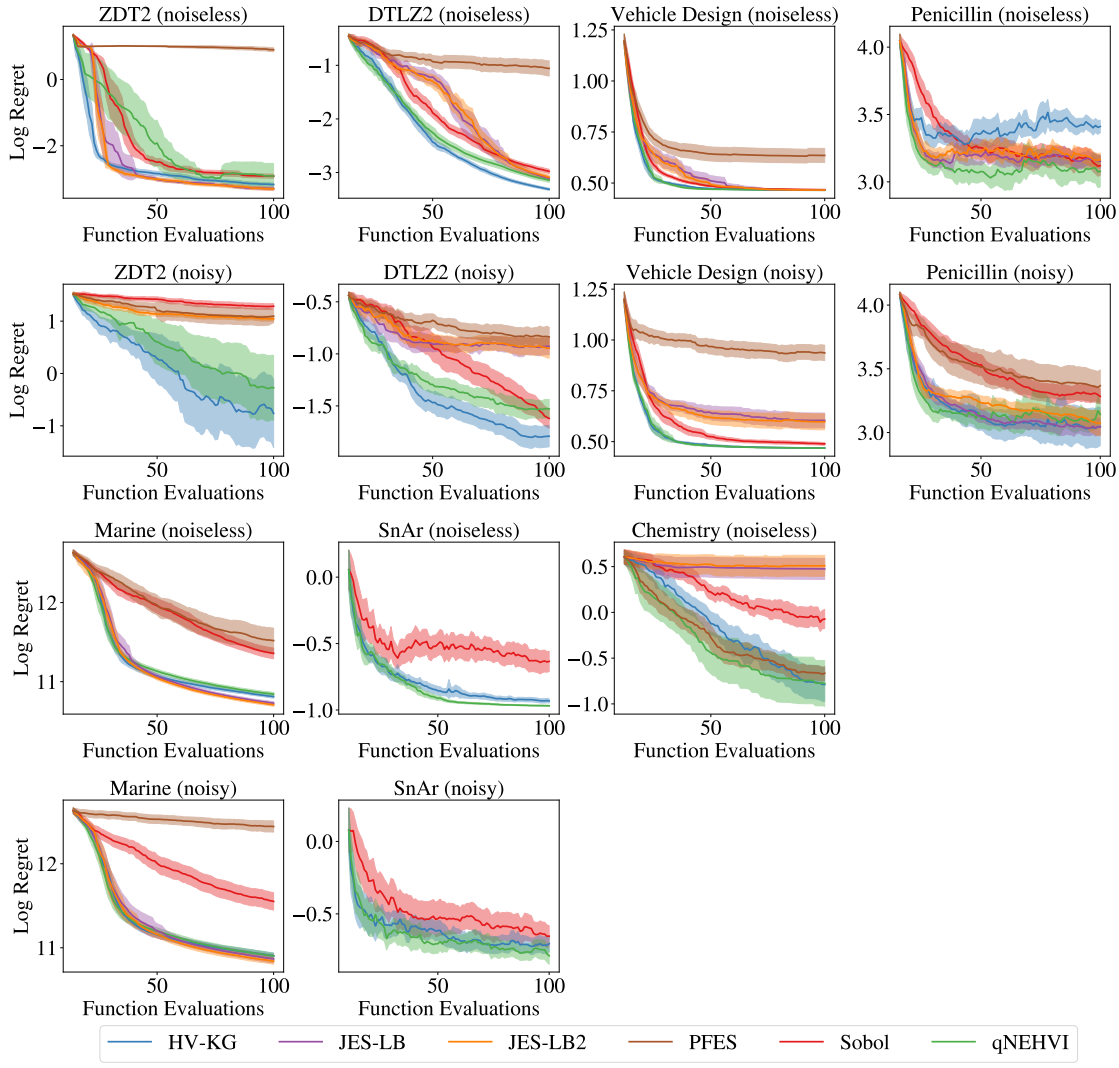

Figure 5.D.1: Sequential ($q = 1$) optimization performance on single fidelity problems.

Parallel MOBO with Complete Information

We evaluate optimization performance using a batch size of $q = 4$ (Figure 5.D.2). We find that HV-KG is a top performer on most problems. Like the sequential case, HV-KG is outperformed by qNEHVI for noiseless Penicilin, and performance is otherwise slightly better than, or not statistically significant from qNEHVI.

5.D.2 Sensitivity with Respect to Pareto Set Size and MC Samples

We evaluate the sensitivity of HV-KG to the Pareto set size N_p and the number of MC samples N and find that HV-KG is quite robust to both as show in

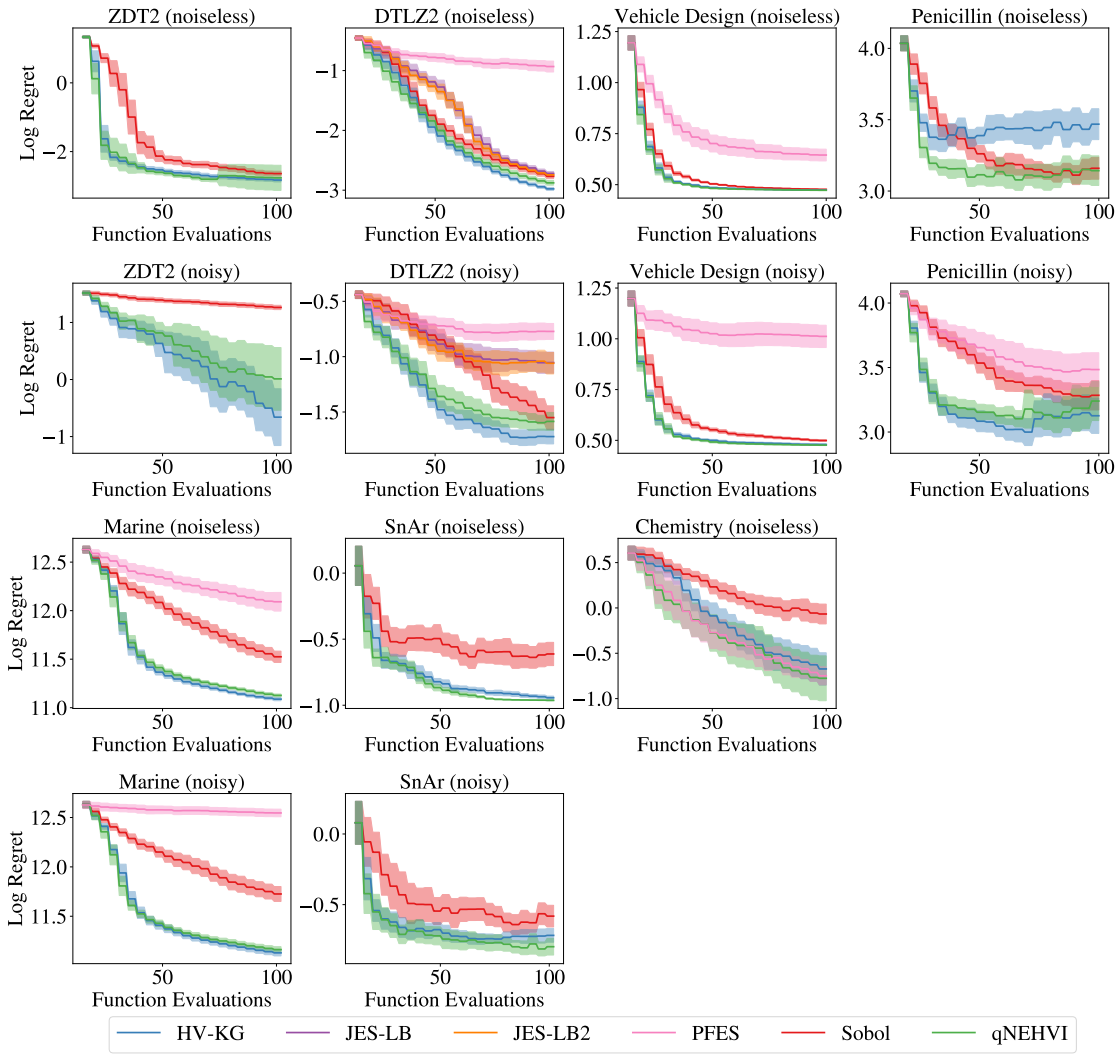

Figure 5.D.2: Parallel ($q = 4$) optimization performance on single fidelity problems. Many PFES and JES-LB(2) runs to failed with numerical errors, and so they are not reported in for some problems.

Figures 5.D.3 and 5.D.4.

5.D.3 Sensitivity to Costs in Competitive Decoupling

In this study, we examine the extent to which CD results are sensitive to the costs used for each objective, which can be particularly relevant when some objectives are more challenging to model than others. To do this, we swap the cost functions such that for ZDT2 and DTLZ2 the two objectives costs 3 and 1, respectively; for Vehicle Design, the objectives have costs 8, 3, and 1 respectively, and for NAS, the objectives have costs 2 and 1 respectively. We observe that decoupled entropy methods works

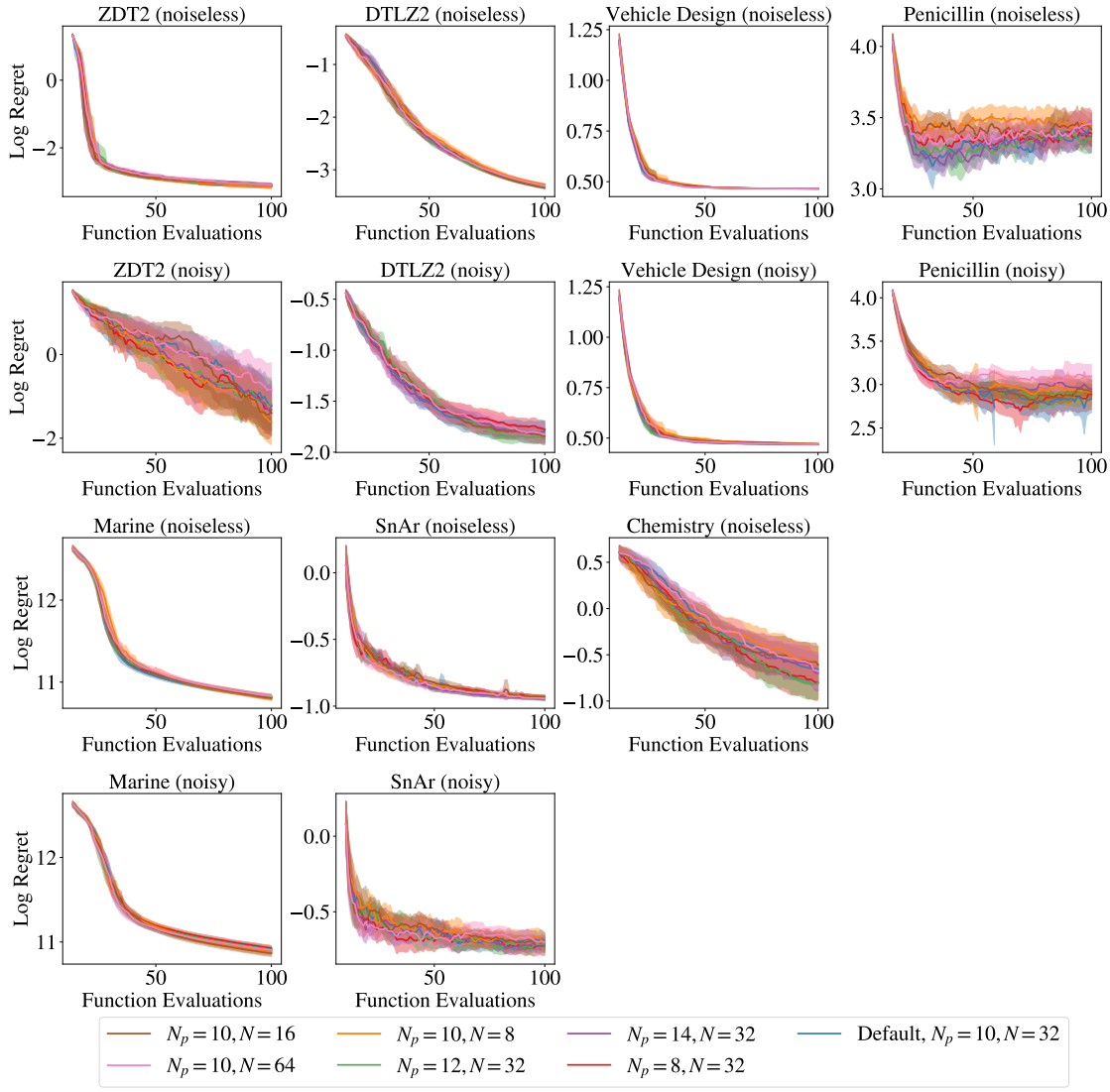

Figure 5.D.3: Sensitivity analysis on single fidelity problems. We do not observe any meaningful differences in performance across multiple problems and over a range of values for either the number of points in the finite Pareto Frontier approximation (N_p) or the number of fantasy samples (N).

significantly better on ZDT2 with the cost functions swapped. We note that the first objective is far simpler than the second objective, and in this case, the first objective is 3 times more expensive. Comparing Figure 5.9.2 in the main text and Figure 5.D.5 here, we find that HV-KG is robust with both cost configurations. In addition, we evaluate which objectives different decoupled algorithms choose to evaluate in the competitive decoupling setting. We observe that the behaviors of JES-LB2 and PFES are far more sensitive to the cost function and that those methods assign

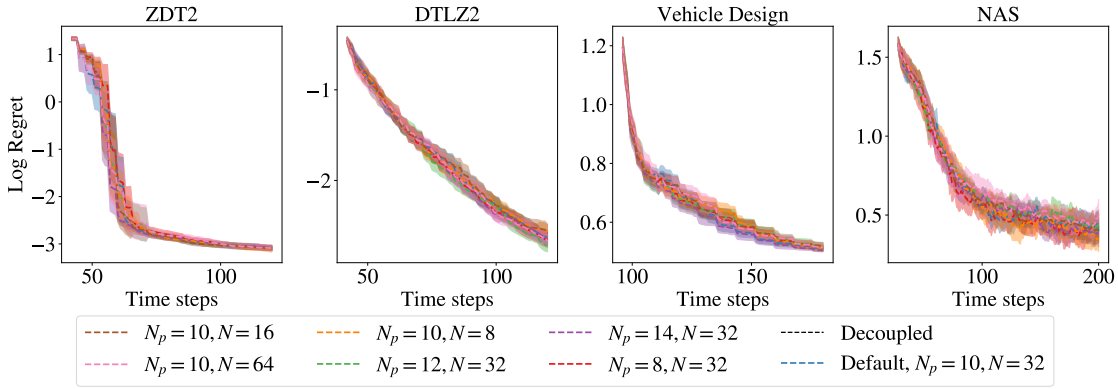

Figure 5.D.4: Sensitivity analysis on NCD problems. We do not observe any meaningful differences in performance across multiple problems and over a range of values for either the number of points in the finite Pareto Frontier approximation (N_p) or the number of fantasy samples (N).

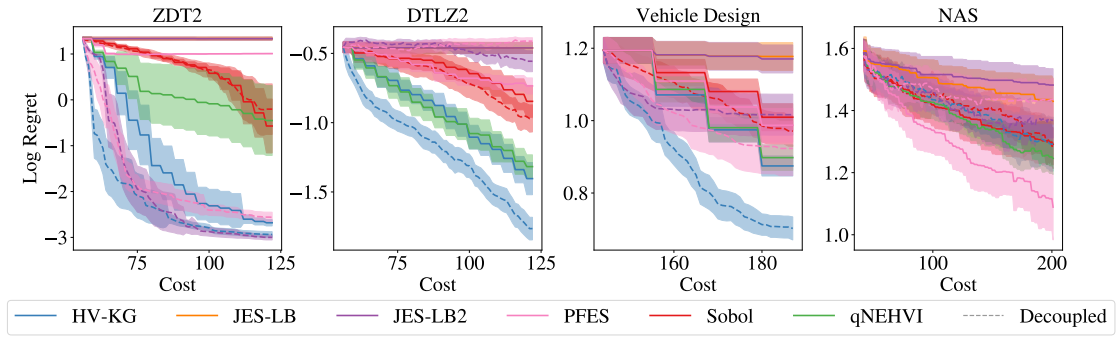

Figure 5.D.5: Competitive decoupling with swapped costs across objectives. Results are qualitatively similar to that of the main text, but the performance of decoupled JES improves for ZDT2, and deteriorates for Vehicle Design.

significantly more samples to lower cost objectives. When the costs are swapped, JES-LB2 and PFES again allocate significantly more evaluations to lower cost objectives, where as the change in HV-KG's behavior is less severe. We suspect the performance of JES-LB2 and PFES is quite sensitive to the choice of cost function.

	ZDT2 (1, 3)	DTLZ2 (1,3)	Vehicle Design (1, 3, 8)	NAS (1, 2)
HV-KG	[34, 29]	[45, 25]	[17, 16, 15]	[182, 159]
JES-LB2	[68, 18]	[78, 14]	[39, 15, 12]	[371, 65]
PFES	-	[66, 18]	[48, 12, 12]	[17, 242]

Table 5.D.1: Number of evaluations of each objective in the competitive decoupling setting.

	ZDT2 (3,1)	DTLZ2 (3,1)	Vehicle Design (8, 3, 1)	NAS (2,1)
HV-KG	[30, 31]	[27, 39]	[14, 18, 21]	[47, 107]
JES-LB2	[21, 57]	[14, 78]	[12, 12, 48]	[26, 148]
PFES	[20, 62]	[30, 32]	[13, 15, 31]	[14, 172]

Table 5.D.2: Number of evaluations of each objective in the competitive decoupling setting with swapped costs across objectives.

	Park	Branin-Currin	Ranking Policy Optimization	Plasma Laser Acceleration
E-HV-KG	336.6 (±45.7)	140.6 (±14.1)	-	-
E-MF-HV-KG	97.3 (±6.0)	74.5 (±2.5)	-	-
HV-KG	15.3 (±1.7)	14.1 (±4.7)	158.6 (±24.4)	55.3 (±9.2)
JES-LB	148.6 (±11.3)	54.8 (±3.1)	68.9 (±4.9)	154.4 (±6.1)
JES-LB2	133.6 (±9.6)	54.1 (±4.4)	70.0 (±3.5)	187.5 (±9.9)
MF-HV-KG	17.9 (±1.4)	16.0 (±0.8)	49.8 (±5.4)	42.7 (±3.1)
MF-Sobol	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)
MOMF	6.5 (±0.4)	5.2 (±0.2)	5.3 (±1.1)	8.6 (±0.6)
PFES	9.3 (±0.2)	11.8 (±1.8)	9.6 (±0.3)	28.0 (±4.0)
Sobol	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)
qNEHVI	2.2 (±0.2)	1.9 (±0.6)	5.0 (±0.3)	3.7 (±0.3)

Table 5.D.3: Acquisition function optimization wall time in seconds on a Tesla V100 SXM2 GPU (16GB RAM) for the multi-fidelity problems. The mean and two standard errors are reported.

5.D.4 Wall Times

We find that candidate generation time with HV-KG is competitive with other methods in the decoupled setting as shown in Tables 5.D.4 and 5.D.5. Notably, HV-KG is significantly faster than the information theoretic alternatives on problems with decoupled evaluations. In the MF setting, MF-HV-KG and HV-KG are slower than alternatives as shown in Table 5.D.3, but MF-HV-KG is also the best performing method with respect to regret.

5.D.5 Wall time of Nested Optimization via Unbiased Estimation

To verify our assertion that nested optimization is far more computationally intensive, we compare the wall time for optimizing HV-KG using the stochastic, unbiased gradient estimator and using our deterministic SAA approach. We use the default stochastic optimization routine in BoTorch [Balandat et al., 2020], which uses Adam [Kingma and Ba, 2014] with a constant learning rate of $\frac{1}{40}$ and an exponential moving average stopping strategy. We use L-BFGS-B to solve the inner optimization

	ZDT2	DTLZ2	Vehicle Design	NAS
HV-KG	16.2 (±1.3)	17.5 (±1.1)	30.6 (±3.8)	12.0 (±0.6)
HV-KG, Decoupled	28.7 (±2.3)	30.8 (±1.7)	55.4 (±6.5)	24.4 (±1.4)
E-HV-KG	203.3 (±9.6)	259.8 (±21.7)	265.4 (±25.1)	15.7 (±0.6)
E-HV-KG, Decoupled	290.0 (±12.7)	479.9 (±30.8)	584.9 (±35.9)	34.8 (±2.0)
JES-LB	100.6 (±8.1)	162.5 (±14.0)	119.9 (±8.9)	44.2 (±0.7)
JES-LB2	110.5 (±9.2)	163.7 (±11.9)	133.1 (±10.7)	43.6 (±0.6)
JES-LB2, Decoupled	172.1 (±13.5)	181.7 (±23.3)	305.4 (±26.4)	44.9 (±0.6)
PFES	15.5 (±1.3)	21.0 (±1.9)	40.5 (±9.0)	16.0 (±0.5)
PFES, Decoupled	-	20.7 (±1.8)	22.1 (±0.8)	17.3 (±0.5)
Sobol	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)	4.2 (±0.3)
Sobol, Decoupled	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)	4.4 (±0.3)
qNEHVI	3.8 (±0.2)	3.9 (±0.2)	4.1 (±0.2)	8.8 (±0.6)

Table 5.D.4: Acquisition function optimization wall time for problems with *competitive decoupling* in seconds on a Tesla V100 SXM2 GPU (16GB RAM). The mean and two standard errors are reported.

	ZDT2	DTLZ2	Vehicle Design	NAS
HV-KG	9.8 (±0.6)	12.9 (±1.7)	22.0 (±2.0)	13.7 (±0.6)
HV-KG, Decoupled	10.7 (±0.5)	12.9 (±0.7)	16.5 (±0.8)	132.4 (±5.4)
E-HV-KG	95.0 (±4.5)	185.4 (±10.1)	149.1 (±10.2)	17.5 (±0.7)
E-HV-KG, Decoupled	116.8 (±5.5)	262.1 (±11.3)	158.1 (±9.1)	-
JES-LB	62.0 (±4.1)	70.3 (±3.6)	68.6 (±3.4)	-
JES-LB2	80.1 (±5.9)	91.3 (±5.8)	91.2 (±6.7)	-
JES-LB2, Decoupled	104.6 (±5.6)	131.8 (±16.1)	177.2 (±11.3)	152.3 (±4.9)
PFES	13.9 (±0.8)	25.4 (±2.2)	38.9 (±3.8)	768.9 (±12.3)
PFES, Decoupled	-	-	349.6 (±8.4)	1421.1 (±36.1)
Sobol	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)	4.1 (±0.3)
Sobol, Decoupled	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)	4.1 (±0.2)
qNEHVI	4.2 (±0.4)	4.4 (±0.3)	4.9 (±0.5)	9.5 (±0.6)

Table 5.D.5: Acquisition function optimization wall time for problems with *non-competitive decoupling* in seconds on a Tesla V100 SXM2 GPU (16GB RAM). The mean and two standard errors are reported.

problem. To select starting points for gradient-based optimization, we sample 8 points from a scrambled Sobol sequence, evaluate HV-KG via solving the inner optimization problem, and use the standard Boltzmann sampling [Duchon et al., 2004] initialization procedure in BoTorch to select a single starting point. We limit the number of quasi-random points to 8 because HV-KG via solving the inner optimization problem is computationally intensive. For the SAA, we use initialization procedure described in Appendix 5.A.2 and we use 1024 quasi-random points to select one starting point (i.e. the current design to select) because

	DTLZ2 (noiseless)	DTLZ2 (noisy)	ZDT2 (noiseless)	ZDT2 (noisy)
E-HV-KG	135.3 (±6.8)	122.2 (±9.8)	69.7 (±3.0)	62.9 (±3.8)
HV-KG	11.3 (±0.2)	11.3 (±0.2)	11.9 (±1.0)	10.1 (±0.6)
JES-LB	98.2 (±9.0)	49.3 (±1.7)	-	44.2 (±1.7)
JES-LB2	133.2 (±3.9)	49.3 (±2.2)	130.8 (±5.9)	42.1 (±1.3)
PFES	33.2 (±1.9)	17.2 (±0.9)	16.0 (±0.7)	14.1 (±0.6)
Sobol	2.8 (±0.0)	2.5 (±0.1)	3.3 (±0.0)	2.3 (±0.0)
qNEHVI	6.2 (±0.1)	6.9 (±0.3)	5.8 (±0.2)	5.5 (±0.2)
	Vehicle Design(noiseless)	Vehicle Design (noisy)	Penicillin (noiseless)	Penicillin (noisy)
E-HV-KG	91.5 (±3.1)	86.4 (±1.8)	218.7 (±14.9)	120.8 (±16.5)
HV-KG	34.3 (±1.4)	29.6 (±0.7)	82.0 (±9.8)	45.4 (±10.1)
JES-LB	167.4 (±9.2)	115.6 (±4.9)	172.2 (±27.8)	83.6 (±5.6)
JES-LB2	210.6 (±10.1)	118.5 (±4.3)	187.7 (±18.4)	82.9 (±4.9)
PFES	37.7 (±2.3)	36.2 (±3.6)	-	37.9 (±2.9)
Sobol	15.8 (±0.5)	12.4 (±0.5)	6.3 (±0.1)	6.6 (±0.3)
qNEHVI	29.7 (±1.3)	27.5 (±1.1)	15.1 (±1.1)	14.7 (±0.8)
	SnAr (noiseless)	SnAr (noisy)	Marine (noiseless)	Marine (noisy)
E-HV-KG	67.9 (±3.5)	127.1 (±10.0)	219.4 (±15.4)	194.8 (±10.3)
HV-KG	24.7 (±0.9)	21.4 (±1.7)	74.9 (±2.6)	66.3 (±1.9)
JES-LB	-	-	245.2 (±14.3)	196.0 (±9.8)
JES-LB2	-	-	274.2 (±10.4)	189.7 (±8.2)
PFES	-	-	83.8 (±7.8)	59.0 (±3.2)
Sobol	13.4 (±1.6)	13.5 (±3.1)	8.0 (±0.3)	7.8 (±0.5)
qNEHVI	19.5 (±0.9)	20.2 (±2.6)	69.9 (±2.5)	76.4 (±3.1)
	Chemistry			
E-HV-KG	47.6 (±2.2)	-	-	-
HV-KG	8.8 (±0.7)	-	-	-
JES-LB	49.6 (±1.9)	-	-	-
JES-LB2	48.6 (±1.2)	-	-	-
PFES	11.9 (±0.4)	-	-	-
Sobol	3.4 (±0.2)	-	-	-
qNEHVI	5.9 (±0.5)	-	-	-

Table 5.D.6: Sequential ($q = 1$) acquisition function optimization wall time in seconds on a Tesla V100 SXM2 GPU (16GB RAM). The mean and two standard errors are reported. PFES and JES-LB(2) failed are missing some values due to numerical errors , which caused the runs to fail.

evaluating HV-KG in a one-shot fashion (i.e. not solving the inner optimization problem to completion for each $\boldsymbol{x}$) is fast. From the starting pint, we use L-BFGS-B to HV-KG in a one-shot fashion. We report results on optimizing HV-KG under these two approaches in Figure 5.D.6 using a GP fit to 14 data points collected from the DTLZ2 ($d = 6, M = 2$) problem [Deb et al., 2002]. We find that using sample average approximation (SAA) determinsitic one-hot optimization finds better candidates and stochastic nested optimization and does so in a fraction of the wall time. It worth noting that we limited the number of quasi-random points to 8 to run this comparison in a reasonable amount of time, but by the time gradient-based optimization starts for the nested stochastic approach, the SAA approach has achieved a higher HV-KG than the stochastic approach will ever reach (on average).

	DTLZ2 (noiseless)	DTLZ2 (noisy)	ZDT2 (noiseless)	ZDT2 (noisy)
HV-KG	43.1 (±2.4)	49.6 (±1.7)	44.5 (±2.0)	35.4 (±1.6)
JES-LB	249.0 (±9.3)	72.1 (±4.0)	-	-
JES-LB2	333.3 (±15.1)	79.2 (±3.9)	-	-
PFES	1420.2 (±29.8)	1224.8 (±23.8)	-	-
Sobol	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)	0.3 (±0.0)
qNEHVI	15.6 (±0.4)	18.9 (±0.6)	7.9 (±0.2)	11.1 (±0.5)
	Vehicle Design(noiseless)	Vehicle Design (noisy)	Penicillin (noiseless)	Penicillin (noisy)
HV-KG	69.5 (±2.5)	64.6 (±1.7)	333.7 (±42.5)	210.8 (±40.4)
PFES	1264.0 (±77.3)	1343.7 (±76.5)	-	1376.4 (±50.7)
Sobol	0.3 (±0.0)	0.2 (±0.0)	0.3 (±0.0)	0.3 (±0.0)
qNEHVI	35.9 (±0.9)	37.1 (±0.9)	35.6 (±1.9)	45.0 (±2.1)
	SnAr (noiseless)	SnAr (noisy)	Marine (noiseless)	Marine (noisy)
HV-KG	45.9 (±1.6)	30.3 (±0.8)	252.3 (±8.4)	225.8 (±7.2)
PFES	-	-	1612.1 (±97.1)	1468.8 (±94.6)
Sobol	0.2 (±0.0)	0.2 (±0.0)	0.2 (±0.0)	0.2 (±0.0)
qNEHVI	11.4 (±0.7)	9.2 (±0.6)	245.4 (±10.4)	264.6 (±10.0)
	Chemistry			
HV-KG	23.6 (±0.6)	-	-	-
PFES	1201.3 (±29.3)	-	-	-
Sobol	0.3 (±0.0)	-	-	-
qNEHVI	5.7 (±0.3)	-	-	-

Table 5.D.7: Batch ($q = 4$) acquisition function optimization wall time in seconds on a Tesla V100 SXM2 GPU (16GB RAM). The mean and two standard errors are reported. Most of the PFES and JES-LB(2) runs to failed with numerical errors.

5.D.6 Fidelity Selection Behavior

In this section, we examine the how different algorithms select fidelities. We examine the fidelity levels which MOMF and MF-HV-KG choose to evaluate at each iteration. We observe that MOMF tends to evaluate many more higher fidelities in early iterations and therefore exhausts its cost budget very quickly. In contrast, MF-HV-KG evaluates many more low-fidelity points early on and therefore collects many more observations (at lower fidelities).

5.E On Pareto Subset Selection

In general, the quality of a finite approximation of a larger (potentially infinite) Pareto frontier is often assessed by additive (and multiplicative) approximation ratios [Bringmann and Friedrich, 2013], which are the minimum added value (and multiplier, respectively) that when applied to all points in the approximate Pareto frontier yield a frontier that is at least a good as all points on the true Pareto frontier. In the bi-objective case, the hypervolume maximizing set enjoys the

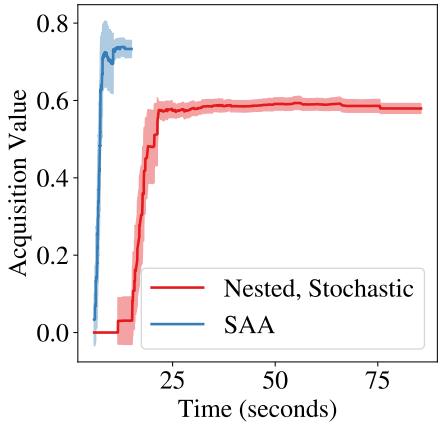

Figure 5.D.6: Acquisition optimization using (i) sample average approximation with deterministic one-shot optimization and (ii) nested optimization with stochastic unbiased gradients. For evaluation at each step of gradient-based optimization, we compute the HV-KG at the current design $\boldsymbol{x}$ by solving the inner optimization problem using L-BFGS-B using 32 (stochastic) fantasy samples. We report the mean and two standard errors of the mean across 20 replications.

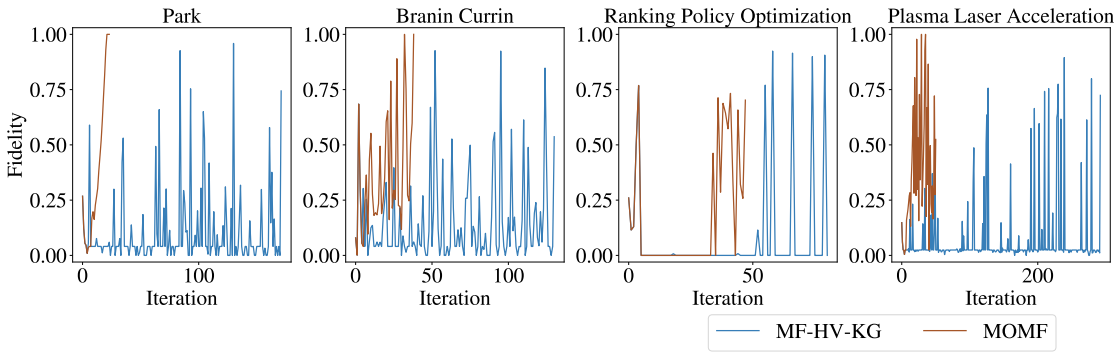

Figure 5.D.7: A comparison of which fidelities each algorithm chooses to query at each iteration.

optimal additive (and multiplicative) approximation ratio(s) asymptotically in N_p [Bringmann and Friedrich, 2013].

Endnote

Clarifications

In Figure 5.2.3, log regret refers to the logarithm of the hypervolume inference regret, which is a commonly used performance metric using multi-objective Bayesian optimization and is defined as in Section 5.9. In all figures in this chapter (including the appendix), we report the mean and 2 standard errors of the mean over 20 replications.

Alternative Approaches

An alternative to multi-objective optimization is preference learning. For example, future work on Astudillo and Frazier [2020] extending the unknown utility formulation to handle partial information using a KG acquisition function would be interesting. In this work however, we focus on learning the entire Pareto frontier, rather than trying to learn the decision-maker's preferences.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Hypervolume Knowledge Gradient: A Lookahead Approach for Multi-Objective Bayesian Optimization with Partial Information
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Hypervolume knowledge gradient: a lookahead approach for multi-objective Bayesian optimization with partial information. *Under review at NeurIPS*, 2023.

Student Confirmation

Student Name:	Samuel Daulton		
Contribution to the Paper	I independently thought of, derived, and implemented this methodology. I ran the experiments for paper, conducted all additional analyses, formulated and proved the theoretical results, and wrote the manuscript. My co-authors played advisory roles.		
Signature	Samuel James Daulton	Date	28 February 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Michael Osborne, Professor of Machine Learning	
Supervisor comments I endorse the description above, which I understand to be correct. Sam indisputably made a substantial contribution to the publication.		
Signature	Date	28 February 2023

This completed form should be included in the thesis, at the end of the relevant chapter.

6

Bayesian Optimization over Discrete and Mixed Spaces via Probabilistic Reparameterization

Contents

6.1 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 leverages a probabilistic surrogate model and an acquisition function (AF) to select promising designs to evaluate. However, maximizing the AF over mixed or high-cardinality discrete search spaces is challenging because standard gradient-based methods cannot be used directly and evaluating the AF at every point in the search space would be computationally prohibitive. 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 original BO policy using

the underlying AF. Moreover, our approach 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. Therefore, as the number 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 a wide range of real-world applications. PR is complementary to (and benefits) recent work and naturally generalizes to settings with multiple objectives and black-box constraints.

6.2 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—to maximize properties such as the oil absorption capacity and mechanical strength [Wang et al., 2020] can involve tuning both discrete ordinal experimental conditions and continuous parameters controlling the composition of the material. For another example, optimizing the structural design of a welded beam can involve tuning the type of metal (categorical), the welding type (binary), and the dimensions of the different components of the beam (discrete ordinals)–resulting in a search space with over 370 million possible designs [Tran et al., 2019]. We consider the scenario where querying the objective function is expensive and sample-efficiency is crucial. In the case of designing the oil sorbent, evaluating the objective function requires manufacturing the material and measuring its properties in a laboratory, requiring significant time and resources.

Bayesian optimization (BO) is a popular technique for sample-efficient black-box optimization, due to its proven performance guarantees in many settings [Srinivas et al., 2010, Berkenkamp et al., 2019] and its strong empirical performance [Frazier, 2018, Turner et al., 2021]. 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] and continuous (or gradient-based) optimization routines cannot be directly applied. Theoretical performance guarantees of BO policies require that the maximizer of the AF is found and selected as the next design to evaluate on the black-box objective function [Srinivas et al., 2010]. 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 propose a technique, probabilistic reparameterization (PR), for maximizing AFs over discrete and mixed spaces by instead optimizing a probabilistic objective (PO): the expectation of the AF over a probability distribution of discrete random variables corresponding to 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 introduced probability 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 [Srinivas et al., 2010]. Although the AF is often non-convex

[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.

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.

6.3 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]

BO leverages (i) a probabilistic surrogate model—typically a Gaussian process (GP) [Rasmussen, 2004]—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) [Jones et al., 1998] and upper confidence bound (UCB) [Srinivas et al., 2010]—the latter of which enjoys no-regret guarantees in certain settings [Srinivas et al., 2010]. 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, it can be optimized numerically. Gradient-based optimization routines are often used to maximize the AF over continuous domains [Garnett, 2023].

[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 In its basic form, BO assumes that the inputs are continuous. However, discrete parameters such as binary, discrete ordinal, and non-ordered categorical parameters 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 [Nguyen et al., 2020a, Ru et al., 2020] and local search [Oh et al., 2019] for discrete domains and interleaved discrete/continuous optimization procedures for mixed domains [Deshwal et al., 2021a, Wan et al., 2021]. A simple and widely-used approach across many popular BO packages [Balandat et al., 2020, The GPyOpt authors, 2016] 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 6.3.1.

Table 6.3.1: Different parameter types, their continuous relaxations, and discretization functions.

Type	Domain	Cont. Relaxation	discretize(·) 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. 6.3.1 (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 [2020] propose discretizing z' before evaluating the AF, but the AF is then non-differentiable with

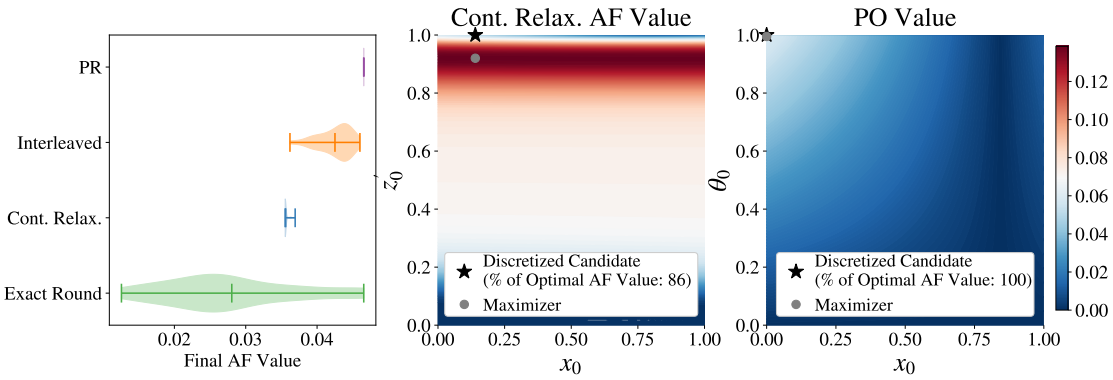

Figure 6.3.1: (**Left**) A comparison of AF optimization using different methods over a mixed search space shows that *PR outperforms alternative methods for AF optimization and has much lower variance across replications.* The violin plots show the distribution of final AF values and the mean. "Cont. 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 [Garrido-Merchán and Hernández-Lobato, 2020]. "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 [Wan et al., 2021]). For each method, the best candidate across 20 restarts is selected (after discretization) and the acquisition value of the resulting feasible candidate is recorded. The AF is expected improvement [Jones et al., 1998]. (**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 6.B 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 a suboptimal AF value when rounded (black star); the resulting candidate only has 86% of the AF value of the true maximizer. The maximum AF value across the feasible search space is shown in white and the red regions indicate that the continuous relaxation overestimates the AF value since it is greater than the maximum AF value of any feasible design. (**Right**) The PO is maximized at the AF unique maximizer within the valid search domain. These contours show that PR avoids the overestimation issue that the naive continuous relaxation suffers from.

respect to the z'. While this improves performance on small search spaces, the response surface has large flat regions after discretizing z', which makes it difficult to optimize the AF. The authors of [Garrido-Merchán and Hernández-Lobato, 2020] propose to approximate the gradients using finite differences, but, empirically, we find that this approach to be leads to sub-optimal AF optimization relative to PR.

6.4 Probabilistic Reparameterization

Algorithm 2 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

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 $\boldsymbol{\theta}$. 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, we define the probabilistic objective (PO):

$$\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})]. \tag{6.1}$$

Algorithm 2 outlines BO with probabilistic reparameterization.

PR allows us to optimize $\boldsymbol{\theta}$ and $\boldsymbol{x}$ over a continuous space to maximize the PO instead of optimizing $\boldsymbol{x}$ and $\boldsymbol{z}$ 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:

$$\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{\boldsymbol{z} \in \mathcal{Z}} p(\boldsymbol{z}|\boldsymbol{\theta}) \alpha(\boldsymbol{x}, \boldsymbol{z}). \tag{6.2}$$

Example distributions for binary, ordinal, and categorical parameters are provided in Table 6.4.1.

Table 6.4.1: Examples of probabilistic reparameterizations for different parameter types. 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	$Z \sim \text{Categorical}(\theta), \theta = (\theta^{(1)}, ..., \theta^{(C)})$	$\theta \in \Delta^{C-1}$

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) and it naturally encodes the ordering of the values. 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

$$\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{z^{(1)}\in\mathcal{Z}^{(1)}} \cdots \sum_{z^{(d_z)}\in\mathcal{Z}^{(d_z)}} \alpha\Big(\boldsymbol{x}, z^{(1)}, ..., z^{(d_z)}\Big) \prod_{i=1}^{d_z} p(z^{(i)}|\theta^{(i)}). \tag{6.3}$$

6.4.1 Analytic Gradients

One important benefit of PR is that the PO in (6.1) 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 6.2:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{\boldsymbol{z}\in\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})\nabla_{\boldsymbol{\theta}} p(\boldsymbol{z}|\boldsymbol{\theta}) \tag{6.4}$$

$$\nabla_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \sum_{\boldsymbol{z}\in\mathcal{Z}} p(\boldsymbol{z}|\boldsymbol{\theta})\nabla_{\boldsymbol{x}} \alpha(\boldsymbol{x}, \boldsymbol{z}) \tag{6.5}$$

This enables optimizing the PO (line 4 of Algorithm 2) efficiently and effectively using gradient-based methods.

6.4.2 Theoretical Properties

In this section, we derive theoretical properties of PR. Proofs are provided in Appendix 6.A. 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 6.4.1 (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 2 outlines BO with probabilistic reparameterization. Importantly, Theorem 6.4.1 states that sampling from the distribution parameterized by a maximizer of the PO yields a maximizer of α, and therefore, Algorithm 2 enjoys the performance guarantees of $\alpha(\cdot)$.

Corollary 6.4.1 (Regret Bounds)**.** *Let $\alpha(\boldsymbol{x}, \boldsymbol{z})$ be an acquisition function over a search space $\mathcal{X} \times \mathcal{Z}$ such that when α is applied as part of a BO strategy that strategy has bounded regret . If the conditions for the regret bounds of that BO strategy using α are satisfied, then Algorithm 2 using α enjoys the same regret bound.*

Examples of BO policies with bounded regret include those based on AFs such as upper confidence bound (UCB) [Srinivas et al., 2010] or Thompson sampling (TS) [Russo and Van Roy, 2014] for single objective optimization, and UCB or TS with Chebyshev [Paria et al., 2020] or hypervolume [Golovin and Zhang, 2020] scalarizations in the multi-objective setting.

Although the BO policy selects a maximizer of α is equivalent to the BO policy in Algorithm 2, 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 better results than directly optimizing α or other common relaxations as shown in Figure 6.3.1(Left), where we compare AF optimization methods on the mixed Rosenbrock test problem (see Appendix 6.B for details).

6.5 Practical Monte Carlo Estimators

6.5.1 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

$$\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x},\boldsymbol{Z})] \approx \frac{1}{N}\sum_{i=1}^{N}\alpha(\boldsymbol{x},\tilde{\boldsymbol{z}}_i), \tag{6.6}$$

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 [Kleijnen and Rubinstein, 1996] (also known as REINFORCE [Williams, 1992] and the likelihood ratio estimator [Glynn, 1990]) because it is simple, scalable, and can be computed using the acquisition values $\{\alpha(\boldsymbol{x},\tilde{\boldsymbol{z}}_i)\}_{i=1}^N$ that are used in the MC estimator of the PO. Many alternative lower variance estimators (e.g. Yin et al. [2020], Yin et al. [2019]) would require many additional AF evaluations (see Mohamed et al. [2020] 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

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x},\boldsymbol{Z})] \approx \frac{1}{N}\sum_{i=1}^{N}\alpha(\boldsymbol{x},\tilde{\boldsymbol{z}}_i)\nabla_{\boldsymbol{\theta}}\log p(\tilde{\boldsymbol{z}}_i|\boldsymbol{\theta}). \tag{6.7}$$

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 6.5.1, which are commonly used for computational convenience and stability in probablistic reparameterization [Yin et al., 2020, Yin et al., 2019], and the solution converges as $\tau \to 0$. 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 6.5.1: Transformations where $\tau \in \mathbb{R}_+$ and $\phi, \theta \in \Theta$.

6.5.2 Variance Reduction in Monte Carlo Gradient Estimation

Although the MC gradient estimator in (6.7) is unbiased, score function gradient estimators can suffer from high variance [Mohamed et al., 2020]. Therefore, we adopt a popular technique for variance reduction where the score function itself is used as a control variate, since its expectation is zero under $p(\boldsymbol{Z}|\boldsymbol{\theta})$ [Mohamed et al., 2020]. Score function estimators with this control variate have been shown to be among the best performing gradient estimators [Mohamed et al., 2020]. Moreover, this technique is simple and merely amounts to subtracting a value β from the acquisition value in the score function estimator in Equation (6.7):

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] \approx \frac{1}{N}\sum_{i=1}^{N}[\alpha(\boldsymbol{x}, \tilde{\boldsymbol{z}}_i) - \beta]\nabla_{\boldsymbol{\theta}} \log p(\tilde{\boldsymbol{z}}_i|\boldsymbol{\theta}). \tag{6.8}$$

The β is commonly known as a baseline and is often taken to be a moving average of the (acquisition) values [Mohamed et al., 2020]. See Appendix 6.B for details on β.

6.5.3 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 [Robbins and Monro, 1951] to arrive at our main convergence result for acquisition optimization.

Theorem 6.5.1 (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 $0 < \sum_{t=1}^{\infty} a_t^2 = A < \infty$ and $\sum_{t=1}^{\infty} a_t = \infty$, 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 $t \to \infty$, $m \to \infty$, and $\tau \to 0$, $(\hat{\boldsymbol{x}}_{t,m}, \hat{\boldsymbol{z}}_{t,m}) \to (\boldsymbol{x}^*, \boldsymbol{z}^*) \in \arg\max_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{X}\times\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})$ in probability.*

The significance of Theorem 6.5.1 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 6.3.1(left) that PR yields strong, stable acquisition optimization with only $m = 20$ starting points, 200 steps, and $\tau = \frac{1}{10}$ (see Appendix 6.F for further discussion) and outperforms alternative optimization approaches.

6.6 Related Work

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

Improving models: Historically, methods leveraging tree-based surrogate models, e.g., SMAC [Hutter et al., 2011] and TPE [Bergstra et al., 2011], have

been popular for optimizing discrete or mixed search spaces. Many recent works have considered alternative surrogate models. BOCS encodes categorical parameters as binary variables and uses Bayesian linear regression with pairwise interactions [Baptista and Poloczek, 2018]. COMBO uses a diffusion kernel on the graph defined by the Cartesian product of discrete parameters [Oh et al., 2019]. MerCBO similarly exploits the combinatorial graph, but with Mercer features and Thompson sampling [Deshwal et al., 2021b]. HyBO extends the diffusion kernels to mixed continuous-discrete spaces [Deshwal et al., 2021a]. 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 restricting them to a discrete set. HyBO enjoys a universal function 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. Moreover, the computational issues of such approaches make it difficult to apply them to multi-objective and constrained optimization. Gryffin [Häse et al., 2021] uses kernel density estimation, but is limited to categorical search spaces. MiVaBO uses a linear combination of basis functions (e.g. pseudo Boolean features [Boros and Hammer, 2002] for discrete parameters) with interaction terms[Daxberger et al., 2020]. MVRSM [Bliek et al., 2021] 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 [2020] 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 continuous relaxation of the discrete parameters and therefore is difficult to optimize. CoCaBO [Ru et al., 2020] samples discrete parameters using a multi-armed bandit and optimizes the continuous parameters conditional upon the sampled discrete parameters. However, CoCaBO's performance degrades as number of discrete configurations increases. Casmopolitan [Wan et al., 2021] 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.

Probabilistic reparameterization: PR has been considered for optimizing discrete parameters in other domains such as reinforcement learning [Williams, 1992] and sparse regression [Yin et al., 2020]. However, PR has not been leveraged for BO. Although the reparameterization trick used by Wilson et al. [2018] is in a similar vein to PR, Wilson et al. [2018] reparameterize an existing multivariate normal random variable in terms of standard normal random variables and then 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.

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 [Maddison et al., 2017, Jang et al., 2017], which admit sample-path gradients. However, samples from Concrete distributions are not discrete and 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 (without discretizing the samples)—e.g., GPs with Hamming distance kernels [Ru et al., 2020]. Another approach for gradient propagation in the deep learning community is to use straight-through gradient estimators (STE) [Bengio et al., 2013], where the gradient of the discretization function with respect to its input is estimated using, for example, an identity function. This approach works well empirically in some cases, these estimators are not well-grounded theoretically. Nevertheless, we discuss and evaluate using STE for AF optimization in Appendix 6.G.

6.7 Experiments

In this section, we provide an empirical evaluation of PR on a suite of synthetic problems and real world applications. For PR, we use stochastic mini-batches of $N = 128$ 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 6.E. We optimize PR using Adam [Kingma and Ba, 2014] with an initial learning rate of $\frac{1}{40}$. We show that PR Adam is generally robust to the choice of learning rate (more so than vanilla stochastic gradient ascent) in the sensitivity analysis in Figure 6.L.2 in Appendix 6.L. We compare PR against two alternative acquisition optimization strategies: using a continuous relaxation (Cont. Relax.) and using exact discretization with approximate gradients (Exact Round) [Garrido-Merchán and Hernández-Lobato, 2020]. These approaches optimize the acquisition function with L-BFGS-B with exact and approximate gradients, respectively. In addition, we compare against two state-of-the-art methods for discrete/mixed BO: a modified version of Casmopolitan [Wan et al., 2021] that additionally supports ordinal variables introduced in Wan et al. [2022] and HyBO [Deshwal et al., 2021a], both of which are shown to outperform the other related works discussed in Section 6.6. In addition, we showcase how PR is complementary to existing methods such as trust region methods [Eriksson et al., 2019]. 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 6.G, we provide comparison of TR methods with alterative optimizers and find that PR is the best optimizer when using TRs on 6 of the 7 benchmark problems. See Appendix 6.B 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 [Ru et al., 2020] for the categorical parameters and Matérn-5/2 kernel for all other parameters.[3] Alternative kernels over different representations of categorical parameters such as one-hot encoded vectors, latent

[3] Cont. Relax. is incompatible with a categorical kernel, so we use a Matérn-5/2 with one-hot encoded categorical parameters.

embeddings [Zhang et al., 2019], and known embeddings (e.g. using fingerprint-based reaction encodings for categorical parameters in chemical reaction optimization [Shields et al., 2021]) are evaluated in Appendix 6.I.

Cont. Relax., Exact Round, PR, and PR + TR use expected improvement [Jones et al., 1998, Gardner et al., 2014] for single objective (constrained problems) and expected hypervolume improvement [Emmerich et al., 2006] for the multi-objective oil sorbent problem (where exact gradients with respect to continuous parameters are computed using auto-differentiation [Daulton et al., 2020]). 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 [Daulton et al., 2022b]. For HyBO, we only run 60 BO iterations on SVM due to the large wall time (see Figure 6.7.2) and only report partial results on Cellular Network due to a singular covariance matrix error. See Appendix 6.B 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 6.B for links), and the implementations of all of other methods are available at `https://github.com/facebookresearch/bo_pr`.

6.7.1 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. [2021]). **Mixed Int F1** is a 16-dimensional variant of the F1 function from Tušar et al. [2019] 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.

6.7.2 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 6.C 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) by 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 [Tran et al., 2019].

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 [Dua and Graff, 2017, Liu et al., 2023]. The 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 [Samal et al., 2022] to maximize a coverage quality metric that is a function of signal power and interference [Maddox et al., 2021] over a geographic region of interest. We use the simulator from Dreifuerst et al. [2021].

Direct Arylation Chemical Synthesis Palladium-catalysed direct arylation has generated significant interest in the pharamceutical development sector[Davies and Morton, 2016]. In this problem, the goal is maximize yield for a direct arylation chemical reaction by tuning 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. [2021] in order to facilitate continuous optimization of

temperature and concentration. In Appendix 6.I, we demonstrate that PR can leverage a kernel over fingerprint-based reaction encodings computed via density functional theory (DFT) for the categorical parameters [Shields et al., 2021].

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 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 [Wang et al., 2020].

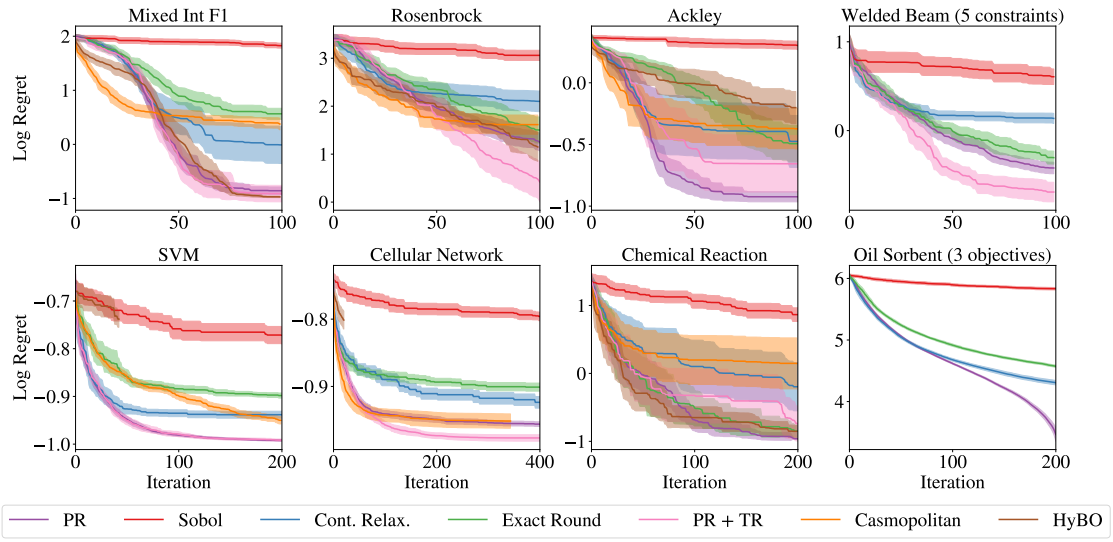

Figure 6.7.1: PR (or PR + TR) consistently outperforms alternatives with respect to log regret.

6.7.3 Results

We find PR consistently delivers strong empirical performance as shown in Figure 6.7.1. *On all benchmark problems, PR (or PR + TR) outperforms all baseline methods (except for Mixed Int F1, where* HyBO *performs comparably).* Figure 6.7.2 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.

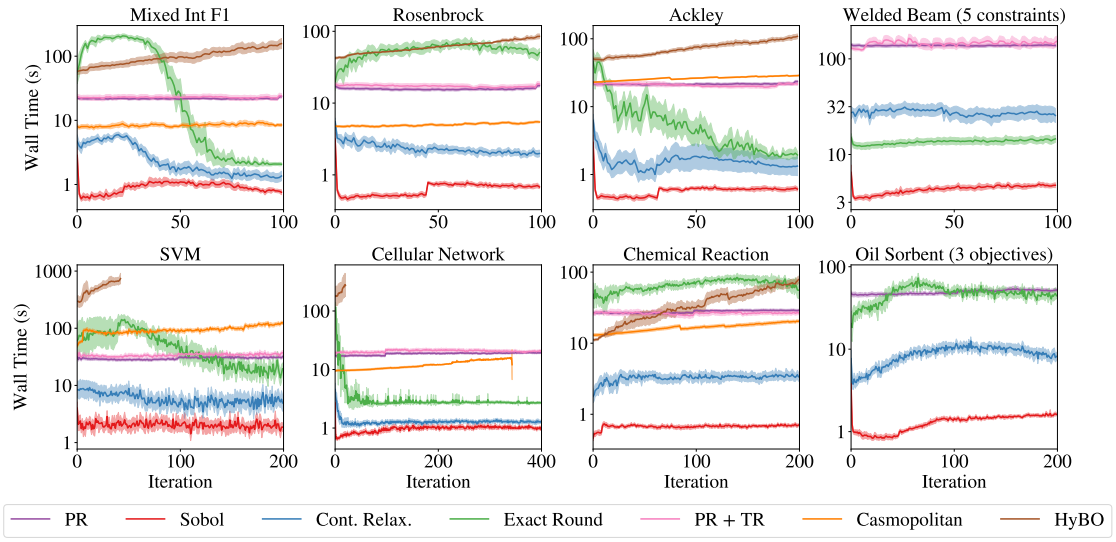

Figure 6.7.2: Wall time for candidate generation at each BO iteration in seconds. Cont. Relax., Exact Round, PR, and PR + TR are run on a single Tesla V100-SXM2-16GB GPU and other methods are run on an Intel Xeon Gold 6252N CPU.

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 scales additively 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 [Daulton et al., 2020, 2021].

6.8 Discussion

The performance and regret properties of BO depend critically on properly maximizing the AF. For problems with discrete features, exhaustively trying all possible combinations of discrete values quickly becomes infeasible as the number of combi-

nations grows. Alternatives such as trying a subset of the possible combinations or resorting to continuous relaxations often leads to a failure to effectively optimize the AF which may result in sub-optimal BO performance. As an alternative, we propose using PR to better optimize the AF, and we demonstrate that PR achieves strong performance on a large number of real-world problems. Our approach is complementary to many other BO extensions, and combines seamlessly with, for example, trust region-based BO and specialized kernels for discrete parameters. One limitation of PR is that it requires computationally-demanding MC integration. However, given that the computation in PR is embarrassingly parallel, it motivates for future research on optimizing AFs on distributed hardware.

Appendix

6.A Theoretical Results and Proofs

6.A.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

$$\tilde{\alpha}(\boldsymbol{x}, p) = \int_{\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) dp(\boldsymbol{z}) = \sum_{\boldsymbol{z} \in \mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) p(\{\boldsymbol{z}\}). \tag{6.9}$$

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

$$\hat{\alpha}(\boldsymbol{x}, \boldsymbol{\theta}) := \tilde{\alpha}(\boldsymbol{x}, \boldsymbol{\varphi}(\boldsymbol{\theta})) = \int_{\mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) dp_{\varphi(\boldsymbol{\theta})}(\boldsymbol{z}) = \sum_{\boldsymbol{z} \in \mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z}) p_{\varphi(\boldsymbol{\theta})}(\{\boldsymbol{z}\}) \tag{6.10}$$

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 $\mathbb{R}^{|Z|}$. 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 6.A.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

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

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 6.A.1. *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 6.A.2. *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 6.A.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 6.A.1.

Clearly, the mappings given in Corollary 6.A.2 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 6.A.1.

Lemma 6.A.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 (6.2),

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

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 6.4.1 (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 6.A.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 6.A.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 6.A.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 6.A.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 6.5.1 (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 $0 < \sum_{t=1}^{\infty} a_t^2 = A < \infty$ and $\sum_{t=1}^{\infty} a_t = \infty$, 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 $t \to \infty$, $m \to \infty$, and $\tau \to 0$, $(\hat{\boldsymbol{x}}_{t,m}, \hat{\boldsymbol{z}}_{t,m}) \to (\boldsymbol{x}^*, \boldsymbol{z}^*) \in \arg\max_{(\boldsymbol{x},\boldsymbol{z}) \in \mathcal{X} \times \mathcal{Z}} \alpha(\boldsymbol{x}, \boldsymbol{z})$ in probability.*

Proof. The binary and categorical mappings in Corollary 6.A.2 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 [Robbins and Monro, 1951]. 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 [Wang et al., 2016a]. From Theorem 6.4.1, the PO and the AF have the same set of maximizers. Hence,

[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.

convergence in probability to a global maximizer of the PO means convergence in probability to a global maximizer of the AF. □

6.B 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 $f^* + 0.1$ 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.

6.B.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 6.7.2).

Ackley. We use an adapted version of the 13-dimensional Ackley function modified from Bliek et al. [2021]. The function is given by:

$$f(\mathbf{x}) = -a \exp\Big(- b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2)}\Big) - \exp\Big(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\Big) + a + \exp(1), \qquad (6.11)$$

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 [Hansen et al., 2019], given by:

$$f(\mathbf{x}) = \sum_{i=1}^{d} (x_i - x_{\text{opt},i})^2 + f_{\text{opt}}, \tag{6.12}$$

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:

$$f(\mathbf{x}) = \Big(\sum_{i=1}^{d-1} \Big(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \Big) \Big), \tag{6.13}$$

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. [2021] (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) [Wang et al., 2017].

6.B.2 Method details

PR, Cont. Relax., Exact Round, PR + TR, and Exact Round + STE. We implemented all of these methods using BoTorch [Balandat et al., 2020], which is available under the MIT license at `https://github.com/pytorch/botorch`. PR and PR + TR use stochastic minibatches of 128 samples and the probabilistic objectives are optimized via Adam using a learning rate of $\frac{1}{40}$. The AFs of Cont. Relax., Exact Round, Exact Round + STE are deterministic and are optimized via L-BFGS-B—Exact Round approximates gradients via finite differences [Garrido-Merchán and Hernández-Lobato, 2020]. All methods use 20 random restarts and are run for a maximum of 200 iterations. We follow the default initialization heuristic in BoTorch [Balandat et al., 2020], 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 Boltzmann sampling [Duchon et al., 2004] 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 [Eriksson et al., 2019] for unconstrained problems and SCBO [Eriksson and Poloczek, 2021] in the presence of outcome constraints, including default trust region update settings.

Casmopolitan: We use the 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. We use the same implementation of interleaved search for the acquisition optimization comparisons.

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 [Deshwal and Doppa, 2021].

6.B.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 [Ru et al., 2020] 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.

6.B.4 Variance Reduction via Control Variates

As discussed in Section 6.5.2, we use moving average baseline for variance reduction. Specifically, the baseline is an exponential moving average with a multiplier of 0.7, where each element is the mean acquisition value across the N MC samples obtained while evaluating the probabilistic objective.

6.B.5 Deterministic Optimization via Sample Average Approximation

Although multi-start stochastic ascent is provably convergent, an alternative optimization approach is to use common random numbers (i.e. a fixed set of base samples) to reduce variance when comparing a stochastic function at different inputs by using the same random numbers. The method of common random numbers 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 [Balandat et al., 2020] and for optimizing risk measures of acquisition functions under random inputs [Daulton et al., 2022a].

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 6.B.1. 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 $Z^{(j)}$, where $j = 1, ..., d_z$ has a corresponding independent base random variable $U^{(j)}$ such that $Z^{(j)} = h(\theta^{(j)}, U^{(j)})$. 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 $\boldsymbol{\theta}$. Under this reparameterization, our probabilistic objective can be written as

$$\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] = \mathbb{E}_{\boldsymbol{U} \sim p(\boldsymbol{U})}[\alpha(\boldsymbol{x}, h(\boldsymbol{\theta}, \boldsymbol{U}))], \tag{6.14}$$

where under the reparameterizations in Table 6.B.1, U is a uniform random variable across the d_z-dimensional unit cube—$P(\boldsymbol{U}) = \text{Uniform}(0, 1)_z^d$. Under this reparameterization we can define our sample average approximation estimator of the probabilistic objective as

$$\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] \approx \frac{1}{N} \sum_{i=1}^{N} \alpha(\boldsymbol{x}, h(\boldsymbol{\theta}, \tilde{\boldsymbol{u}}_i)). \tag{6.15}$$

Our sample average approximation estimator of the gradient of the probabilistic objective with respect to $\boldsymbol{\theta}$ is given by

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})}[\alpha(\boldsymbol{x}, \boldsymbol{Z})] \approx \frac{1}{N} \sum_{i=1}^{N} \alpha(\boldsymbol{x}, h(\boldsymbol{\theta}, \tilde{\boldsymbol{u}}_i)) \nabla_{\boldsymbol{\theta}} \log p(h(\boldsymbol{\theta}, \tilde{\boldsymbol{u}}_i)|\boldsymbol{\theta}). \tag{6.16}$$

Sample average approximation estimators are deterministic and biased conditional on the selection of base samples. However, the reparameterizations in Table 6.B.1 create discontinuities in the PO, and the number of discontinuities increases with the number of MC samples. Nevertheless, we find that optimizing the PO using L-BFGS-B delivers strong performance on the benchmark problems and we compare against stochastic optimization in Figures 6.L.2 and 6.L.1. As in the stochastic case, we reduce the variance further by leveraging quasi-MC sampling [Owen, 2003] instead of i.i.d. sampling.

Table 6.B.1: 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	$Z \sim \text{Categorical}(\boldsymbol{\theta})$	$h(\theta, U) = \min(\arg\max_{i=0}^{C-1} \mathbb{1}(U < \sum_{c}^{i} \theta^{(c)}))$

6.C 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. [2014] 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 [Emmerich et al., 2006] for a review of multi-objective optimization.

6.D 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 6.D.1 and 6.D.2 we compare PR (optimized with Adam using stochastic mini-batches of 128 MC samples) and analytic PR (optimized with L-BFGS-B) 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.

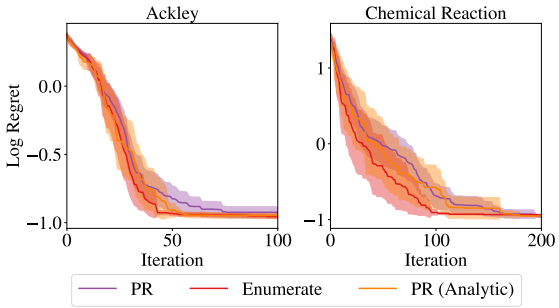

Figure 6.D.1: A comparison with an enumeration (gold standard) with respect to log regret.

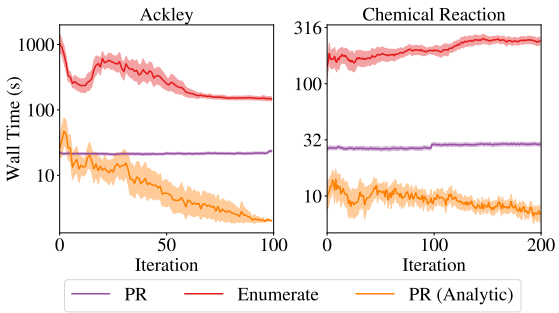

Figure 6.D.2: A comparison with enumeration with respect to wall time.

6.E Analysis of MC sampling in Probabilistic Reparameterization

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 6.E.1). Run time is linear with respect to MC samples, and so substantial compute savings are possible when fewer MC samples are used (Figure 6.E.2). 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 6.E.3, although wall time is considerably faster for fewer 64 or less MC samples as shown in Figure 6.E.4.

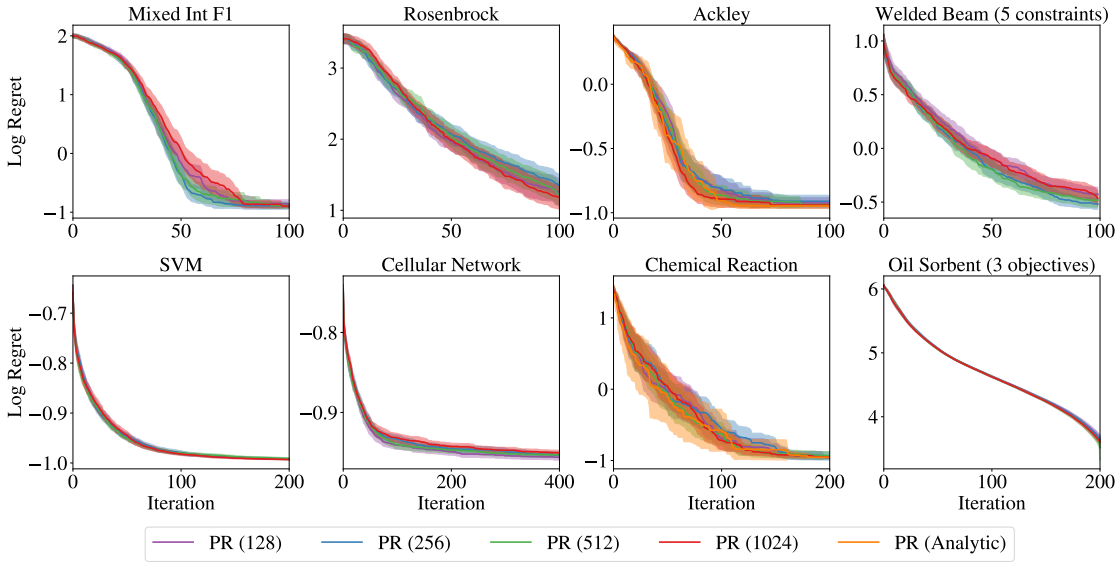

Figure 6.E.1: A sensitivity analysis of the optimization performance of PR with respect to the number of MC samples. We find that PR is robust to the number of MC samples, and that the performance of MC PR matches that of analytic PR.

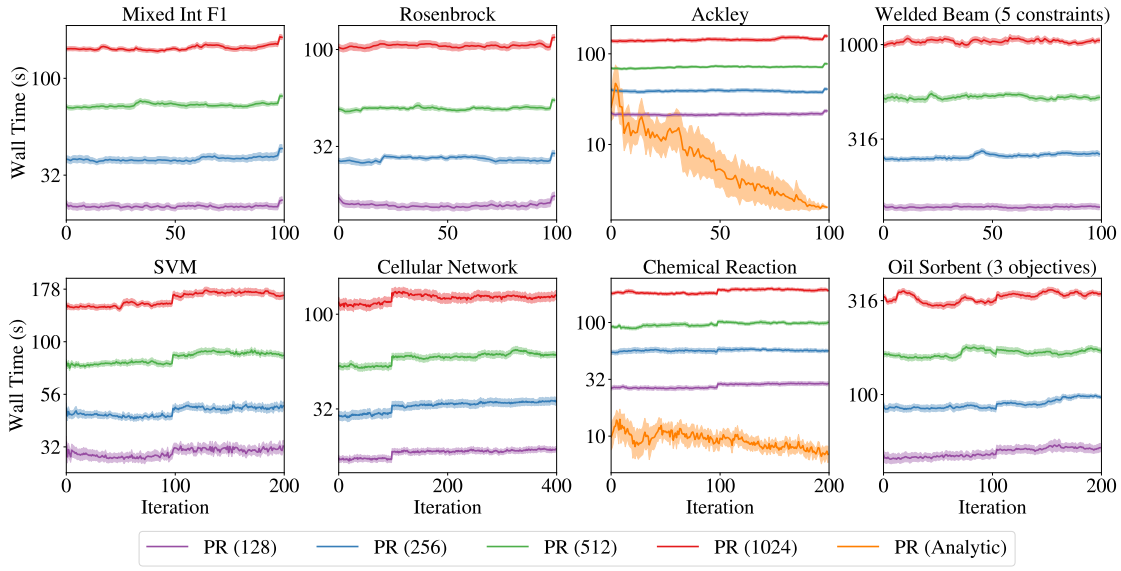

Figure 6.E.2: A sensitivity analysis of the wall time of PR with respect to the number of MC samples. We observe that wall time scales linearly with the number of MC samples, which is expected since we compute PR in $\frac{N}{32}$ chunks to avoid overflowing GPU memory.

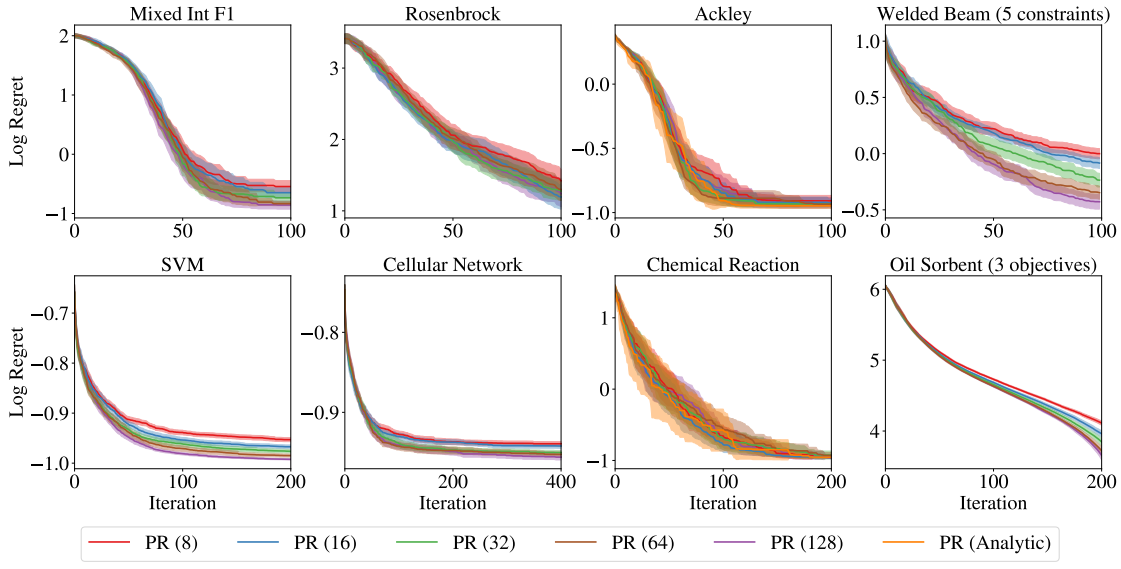

Figure 6.E.3: A sensitivity analysis of the optimization performance of PR with respect to a small number of MC samples (with samples between 8 and 64). Performance degrades slightly when few samples are used.

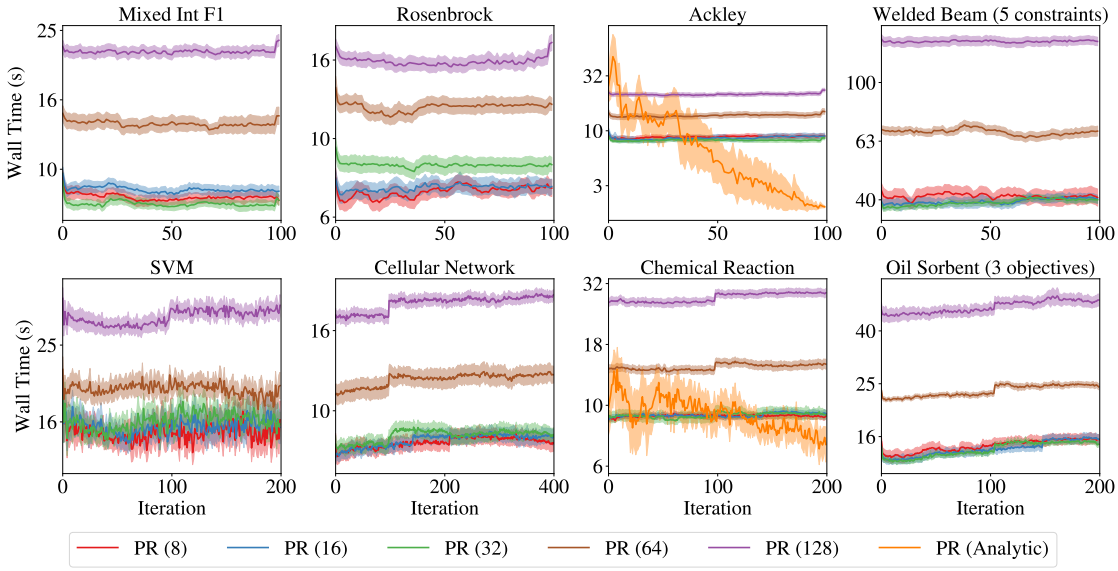

Figure 6.E.4: A sensitivity analysis of the wall time of PR with respect to the number of MC samples (with samples between 8 and 64). We observe that wall time scales linearly with the number of MC samples, which is expected since we compute PR in $\frac{N}{32}$ chunks to avoid overflowing GPU memory.

Evaluation of Approximation Error in MC Sampling

We examine the MC approximation error relative to analytic PR on the chemical reaction and ackley problems. The results in Figure 6.E.5 show the mean absolute percentage error (MAPE)

$$\frac{100}{|X_{\text{discrete}}|} \cdot \sum_{\boldsymbol{x} \in X_{\text{discrete}}, \boldsymbol{\theta} \in \Theta_{\text{discrete}}} \frac{\mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})} \alpha(\boldsymbol{x}, \boldsymbol{Z}) - \frac{1}{N} \sum_{i=1}^{N} \alpha(\boldsymbol{x}, \tilde{\boldsymbol{z}}_i)}{\max_{\boldsymbol{x} \in X_{\text{discrete}}, \boldsymbol{\theta} \in \Theta_{\text{discrete}}} \mathbb{E}_{\boldsymbol{Z} \sim p(\boldsymbol{Z}|\boldsymbol{\theta})} \alpha(\boldsymbol{x}, \boldsymbol{Z})}$$

evaluated over a random set of $|X_{\text{discrete}}| = |\Theta_{\text{discrete}}| 10,000$ points from $\mathcal{X} \times \times$ (the sampled sets are denoted $X_{\text{discrete}}, \Theta_{\text{discrete}}$). We observe a rapid reduction in MAPE as we increase the number of samples. With 1024 samples, MAPE is 0.055% (+/- 0.0002 %) over 20 replications (different MC samples in PR) on the chemical reaction problem and MAPE is 0.018% (+/- 0.0003 %) on the ackley problem.

With 128 samples, MAPE is 0.282% (+/- 0.0029 %) over 20 replications (different MC samples in PR) on the chemical reaction problem and MAPE is 0.052% (+/- 0.0021 %) on the ackley problem.

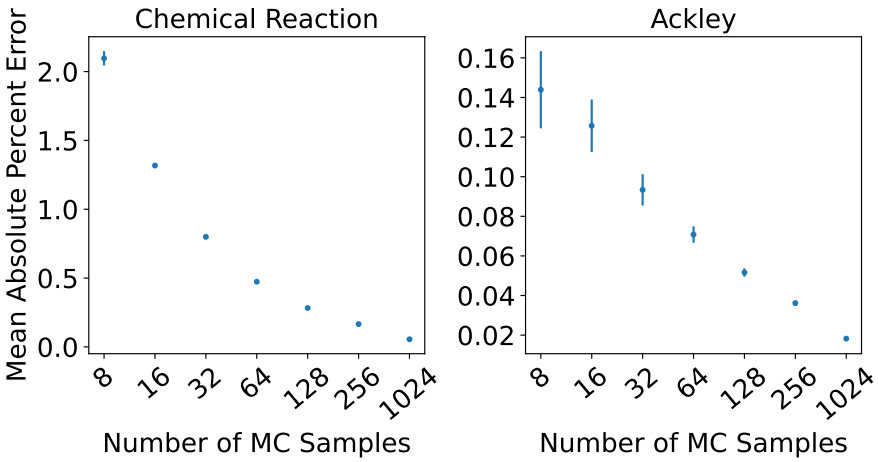

Figure 6.E.5: An evaluation of the mean absolute percentage error for the MC estimator of PR (relative to analytic PR).

6.F Effect of temperature parameter 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 as shown in Figure 6.F.1.

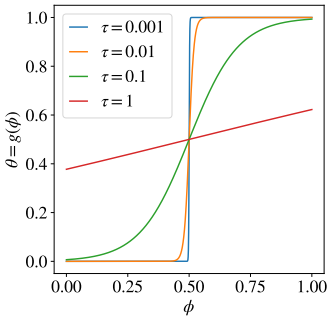

Figure 6.F.1: A comparison of the reparameterization of θ under various choices of τ. We observe that $\tau = 0.1$ provides a reasonable balance between retaining non-zero gradients of $g(\phi)$ with respect to ϕ and allowing θ to become close to 0 or 1.

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.

6.G Alternative methods

6.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) [Bengio et al., 2013]. 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 6.G.1) with fast wall times (Figure 6.G.2), but does not quite match the optimization performance of PR on several benchmark problems.

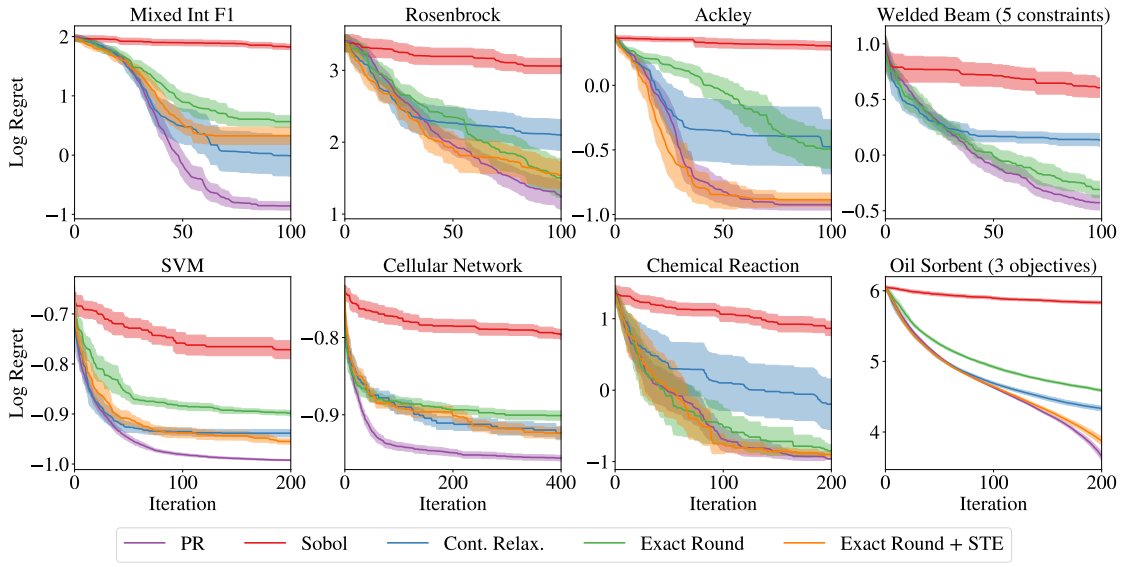

Figure 6.G.1: A comparison of exact rounding with straight-through gradient estimators versus other acquisition optimization strategies. Log regret on each problem. We report log hypervolume regret for Oil Sorbent and report the log regret of the best feasible objective for Welded beam.

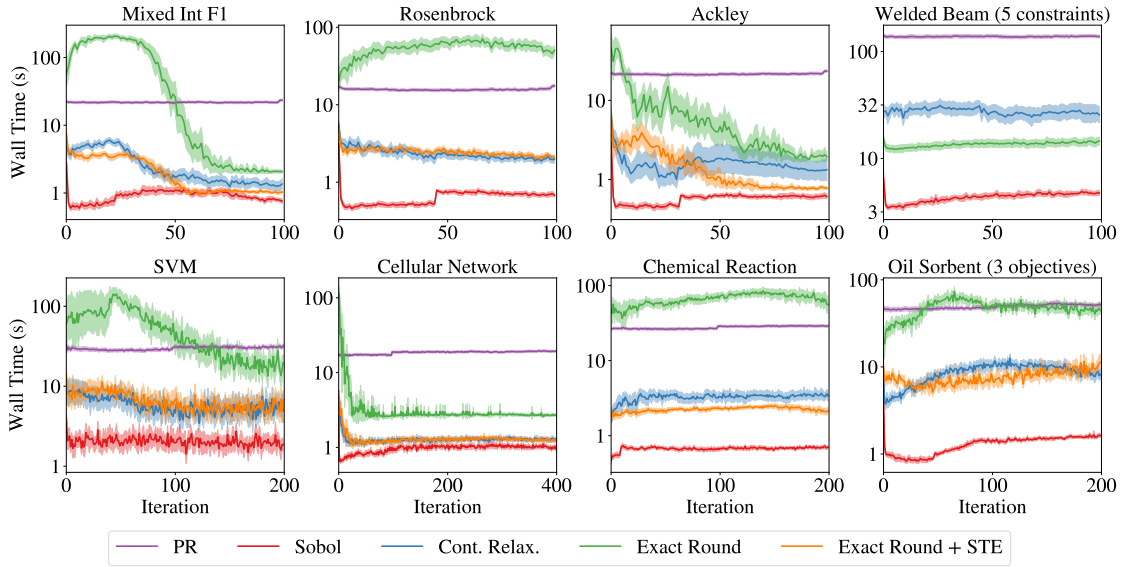

Figure 6.G.2: A comparison of wall times of exact rounding with straight-through gradient estimators versus other acquisition optimization strategies.

6.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 6.G.3 show that PR is a consistent best optimizer using TRs, but that STEs work quite well with TRs in many scenarios.

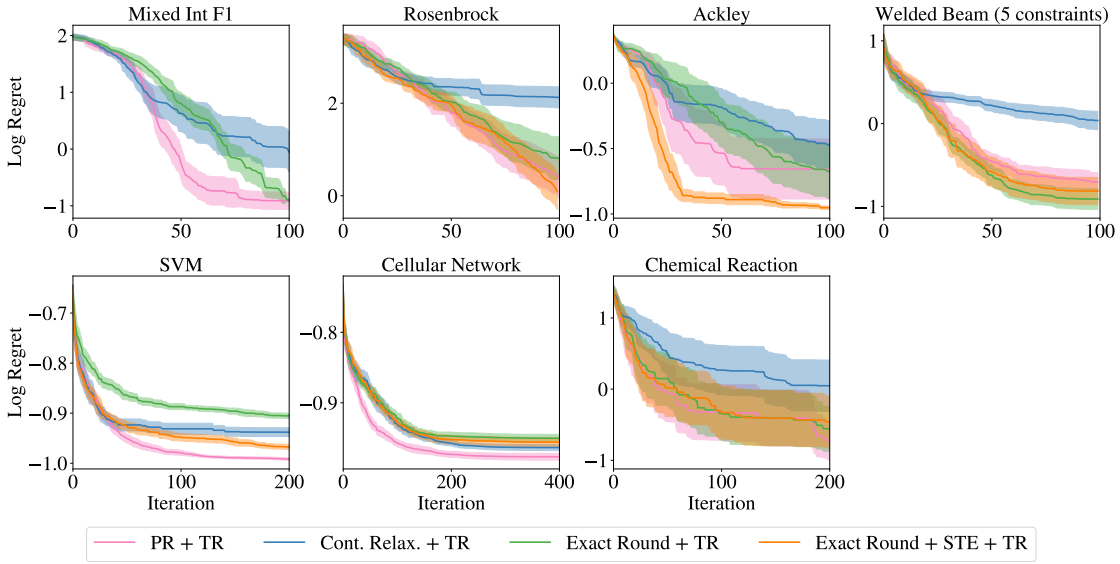

Figure 6.G.3: A comparison of TR methods with different acquisition optimization strategies. Log regret on each problem. We report log hypervolume regret for Oil Sorbent and report the log regret of the best feasible objective for Welded beam.

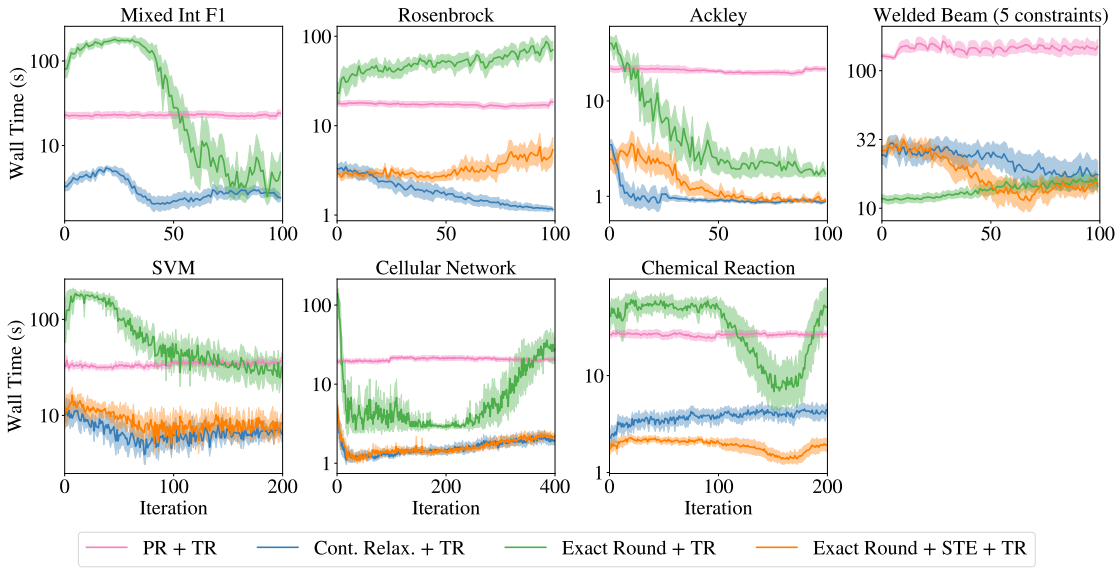

Figure 6.G.4: A comparison of wall times of TR methods with different acquisition optimization strategies.

6.H Acquisition Function Optimization at a Given Wall Time Budget

In Figure 6.H.1, 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.

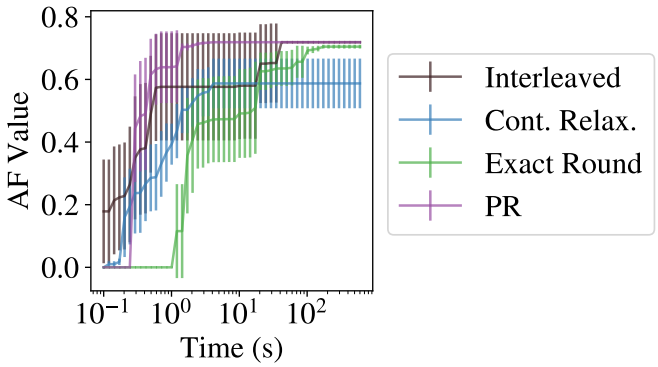

Figure 6.H.1: A comparison of methods for optimizing acquisition functions at a given wall time budget.

6.I 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) a Categorical kernel (default) versus with a Matérn-5/2 kernel with either (b) one-hot encoded categoricals, (c) a latent embedding kernel [Zhang et al., 2019], or known embeddings based on density functional theory (DFT) [Shields et al., 2021]. For the latent embedding kernel, we follow Pelamatti et al. [2021] 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. For the kernel over DFT embeddings, we use the DFT embeddings for the direct arylation dataset from Shields et al. [2021], which are available at `https://github.com/b-shields/edbo`. It is worth noting that in the Chemical Reaction problem, the black-box objective is a GP surrogate model with a Categorical kernel that is fit to the direct arylation dataset. The purpose of this section is demonstrate that PR is agnostic to the choice of kernel over discrete parameters. Because the Chemical Reaction problem is based on a GP surrogate, we do not draw conclusions about which choice of kernel is best suited for modeling the true, unknown underlying Chemical Reaction yield function.

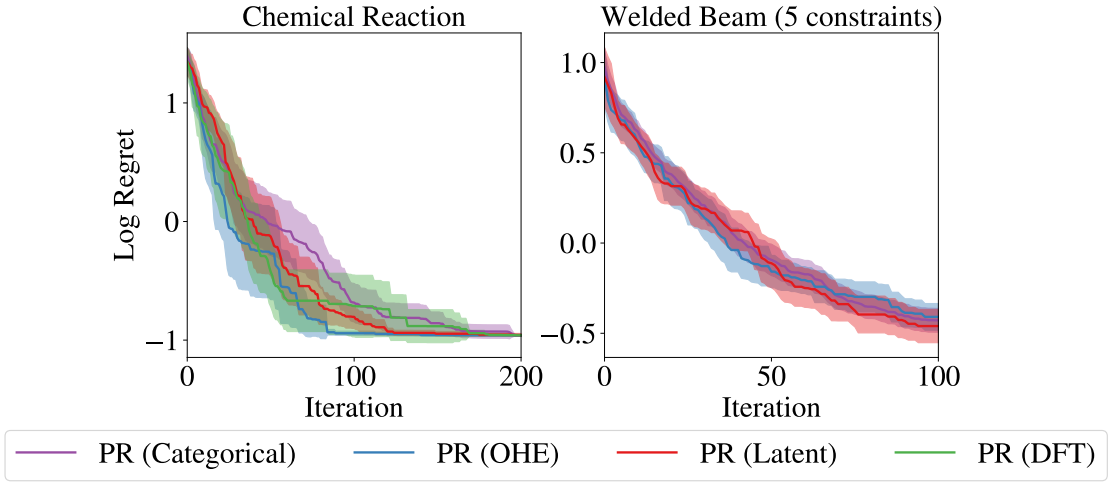

Figure 6.I.1: A comparison of different kernels over categorical parameters. Left: Welded beam has one categorical parameter, metal type (4 levels). Right: Chemical reaction has three categorical parameters, solvent, base, and ligand (with 4, 12, and 4 levels, respectively).

6.J 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. [2015b]. Although UCB comes enjoys bounded regret [Srinivas et al., 2010], we find empirically that EI works better on most problems.

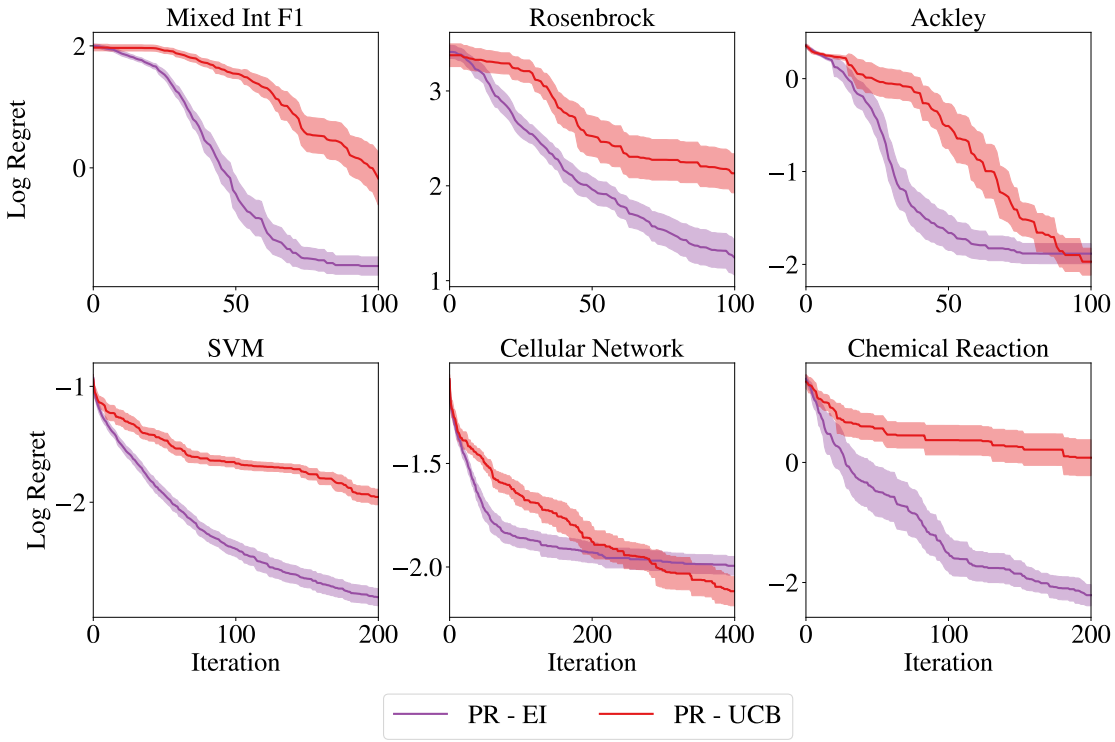

Figure 6.J.1: A comparison of expected improvement (EI) and upper confidence bound (UCB) acquisition functions with PR.

6.K 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.

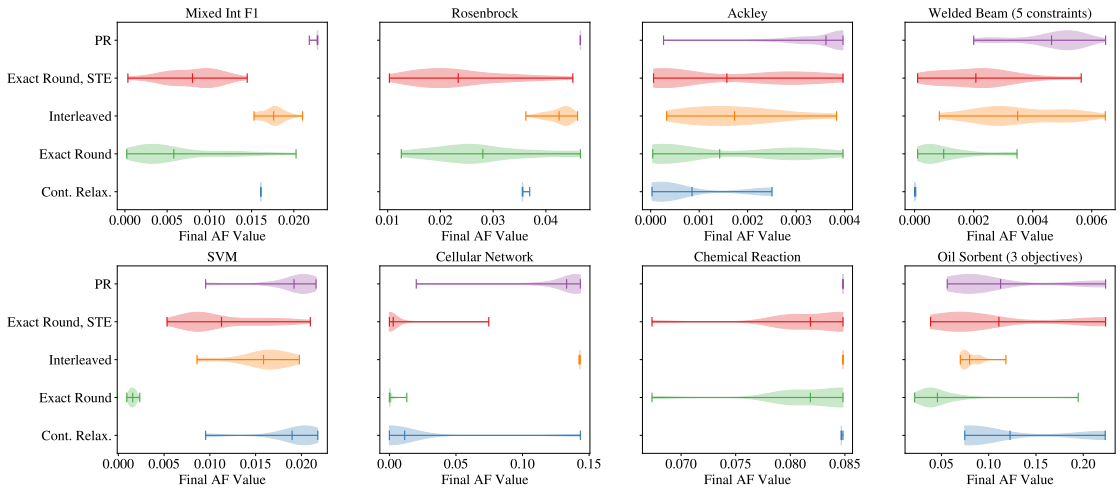

Figure 6.K.1: A comparison of methods for optimizing acquisition functions.

6.L Stochastic vs Deterministic Optimization

We compare optimizing PR with stochastic and deterministic optimization methods. For stochastic optimizers, we compare stochastic gradient ascent (SGA) and Adam with various initial learning rates. For SGA, the learning rate is decayed each time step t by multiplying the initial learning rate by $t^{-0.7}$ and for Adam a fixed learning rate is used. For stochastic optimizers, the MC estimators of PR and its gradient stochastic mini-batches of $N = 128$ MC samples are used. For deterministic optimization, base samples are kept fixed. All routines are run for a maximum of 200 iterations. In Figure 6.L.1, we observe that Adam is more robust to the choice of learning rate than SGA and generally is the best performing method. Furthermore, Adam consistently performs better than deterministic optimization. We compare Adam with a learning rate of $\frac{1}{40}$ against L-BFGS-B in Figure 6.L.2.

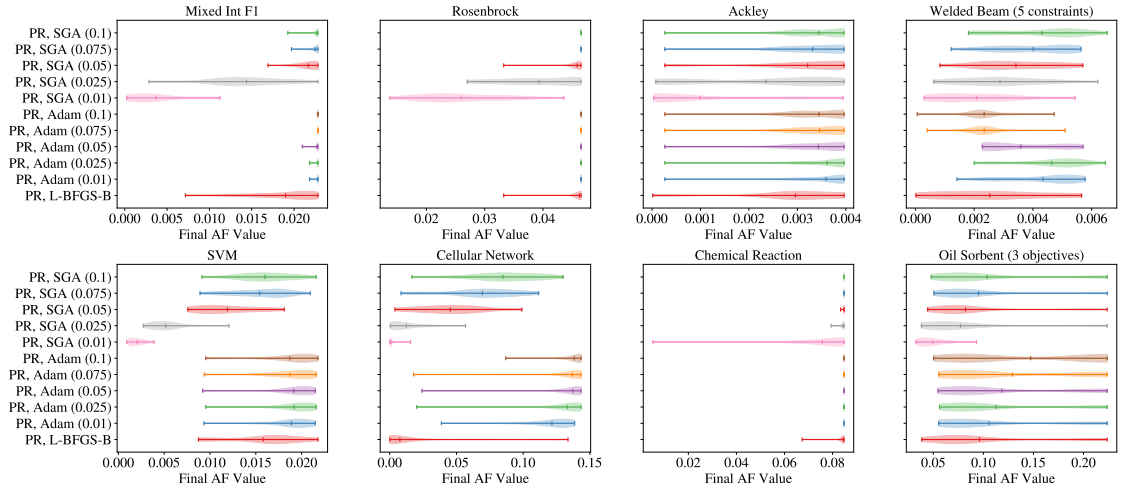

Figure 6.L.1: A comparison of PR using stochastic and deterministic optimization methods. The initial learning rate for stochastic gradient ascent is given in parentheses.

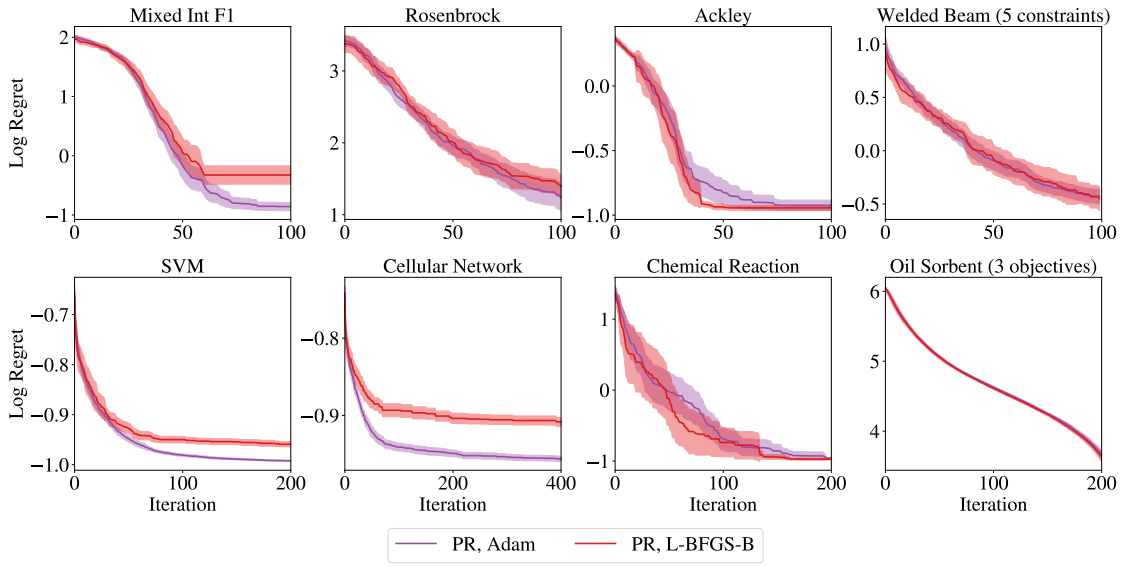

Figure 6.L.2: A comparison of optimizing the PO using deterministic estimation (via SAA) and optimization versus stochastic estimation and optimization.

6.M Comparison with an Evolutionary Algorithm

In Figures 6.M.1 and 6.M.1, we compare against the evolutionary algorithm PortfolioDiscreteOnePlusOne, which is the recommended algorithm for discrete and mixed search spaces in the Nevergrad package [Rapin and Teytaud, 2018]. We find that PR significantly outperforms this baseline by a large margin with respect to log regret, but is slower than the evolutionary algorithm with respect to wall time.

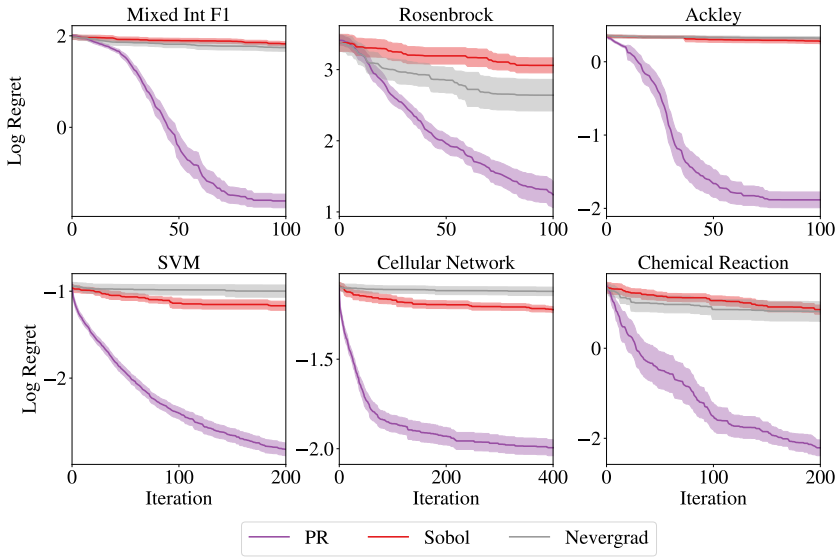

Figure 6.M.1: A comparison with an evolutionary algorithm with respect to log regret.

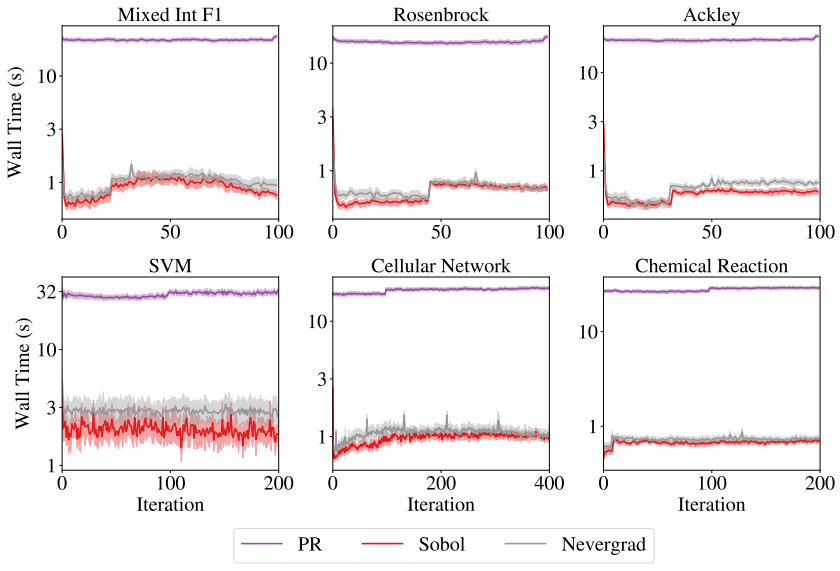

Figure 6.M.2: A comparison with an evolutionary algorithm with respect to wall time.

Endnote

Clarifications

In Section 6.5, $\boldsymbol{\phi}$ is used without proper introduction. $\boldsymbol{\phi}$ is the governing parameter that is transformed into $\boldsymbol{\theta}$ using the transformations in Table 6.5.1.

In Section 6.A, we define $\tilde{\alpha} : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$, but it should be defined as $\tilde{\alpha} : \mathcal{X} \times \mathcal{P}_{\mathcal{Z}} \to \mathbb{R}$.

In Lemma 6.A.1, we assume Θ is compact (as stated in the proof), but we do not explicitly state this in the Lemma itself. Furthermore, our proof requires the additional assumption that Θ contains the parameter $\boldsymbol{\theta}$ that provides support exclusively on a maximizer $\boldsymbol{z}^*$ of α (i.e. $\boldsymbol{x}^*, \boldsymbol{z}^* \in \arg\max_{\boldsymbol{x},\boldsymbol{z}} \alpha(\boldsymbol{x}, \boldsymbol{z})$). That is, Θ contains the parameter $\boldsymbol{\theta}$ such that $p_{\varphi(\boldsymbol{\theta})}(\boldsymbol{z} = \boldsymbol{z}^*) = 1$. In addition, in the proof of Lemma 6.A.1, we state that since both $\mathcal{X}$ and Θ are compact, $\mathcal{J}^*$ exists, but the important part that we do not explicitly state is that $\mathcal{J}^*$ is non-empty.

Unless otherwise noted, we report the mean and two standard errors of the mean over 20 replications in all performance figures in this chapter.

In Section 6.E, "ackley" should be capitalized.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Bayesian Optimization over Discrete and Mixed Spaces via Probabilistic Reparameterization
Publication Status	☑Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Samuel Daulton, Xingchen Wan, David Eriksson, Maximilian Balandat, Michael A. Osborne, and Eytan Bakshy. Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. In *Advances in Neural Information Processing Systems 35*, 2022.

Student Confirmation

Student Name:	Samuel Daulton		
Contribution to the Paper	I independently thought of, derived, and implemented this methodology. I ran the experiments for paper and conducted all additional analyses. Xingchen assisted with running baselines in external codebases, David assisted with modeling and trust region approaches, and the other co-authors played advisory roles.		
Signature	Samuel James Daulton	Date	28 February 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title:	Michael Osborne, Professor of Machine Learning		
Supervisor comments	I endorse the description above, which I understand to be correct. Sam indisputably made a substantial contribution to the publication.		
Signature		Date	28 February 2023

This completed form should be included in the thesis, at the end of the relevant chapter.

7 Conclusion

Contents

7.1 Discussion

This thesis considers scenarios where sample-efficient optimization is challenging and proposes new methods for enhancing Bayesian optimization in each setting. Each chapter focuses on a different adverse scenario.

In Chapter 3, we address the case where the decision maker seeks to optimize multiple objectives under input noise. Although robust Bayesian optimization of risk measures that provide probabilistic guarantees has been considered in the single objective case [Cakmak et al., 2020], it has received less attention in the multi-objective realm. We build upon prior work on a multi-variate value-at-risk [Prèkopa, 2012] and formulate robust multi-objective optimization through this lens. We then propose a simple, practical, and efficient algorithm for robust multi-objective Bayesian optimization based on our theoretical insight regarding approximating the multi-variate value-at-risk set via random scalarizations.

In Chapter 4, we consider optimizing multiple competing objectives over high-dimensional search spaces. Although high-dimensional search spaces are difficult for standard Bayesian optimization methods, there has a been a significant body of research on improving Bayesian optimization performance when there are many tunable parameters [Wang et al., 2016b, Eriksson et al., 2019, Letham et al., 2020, Eriksson and Jankowiak, 2021]. We build upon the core insight in the TuRBO method [Eriksson et al., 2019]—namely, that Bayesian optimization in local trust regions is an effective global optimization strategy in high-dimensions—and propose a new algorithm MORBO for efficiently identifying the Pareto frontier using a collaborative, coordinated multi-trust-region approach. We demonstrate that this improves performance on daunting high-dimensional, relatively high-throughput real-world optimization problems.

Chapter 5 presents a general framework for multi-objective Bayesian optimization with partial information, such as (i) only observing a subset of the objectives for each design or (ii) observing one or more objectives at lower-fidelities than the true target fidelity. We highlight several real-world scenarios where not only do we observe partial information, but the optimization algorithm can choose what partial information to observe for deach design (e.g. the fidelity level or the objectives to evaluate). Although there have been some limited attempts to address these problems, our method is the first general framework to do so. Starting from a Bayesian decision theoretic perspective on Pareto set selection (how we will determine the optimal designs), we construct a one-step Bayes optimal acquisition function for hypervolume maximization under our decision theoretic framework. We propose an efficient estimator of our acquisition function using sample average approximation and show that our method yields improved optimization performance in a wide swath of settings.

In Chapter 6, we consider the case where the search space is combinatorial (consists purely of discrete parameters(or mixed (contains both continuous and discrete tunable parameters). Although this problem setting has received significant

attention, we reconsider the problem through the lens of probabilistic reparameterization, where the discrete parameters reparameterized by discrete random variables with probability distributions governed by continuous parameters. We prove that maximizing the expected value of the acquisition function under these probabilistic reparameterization yields an equivalent Bayesian optimization policy to optimizing the underlying acquisition function. Furthermore, we propose both a Monte Carlo estimator of the acquisition function and a stochastic unbiased gradient estimator for efficient acquisition optimization. We show that this improves optimization performance in both single and multi-objective settings.

7.2 Future Work

Although this thesis tackles several challenging settings for Bayesian optimization by proposing new methods, there are other alternative methods that could be used instead and could be interesting to explore in future work.

For example, MVaR is but one optimization goal that could be used for robust multi-objective optimization as discussed in the endnote of Chapter 3. Considering robustness with respect to a scalarization of the objectives would be a more direct and likely better approach when the scalarization is known a priori. In the case that the scalarization is unknown, incorporating preference learning to learn the unknown utility function within the robust optimization setting could also yield improve sample efficiency.

In Chapter 4, we made design decisions for the MORBO algorithm with practical applications in mind. For example, we chose to use a point's hypervolume contribution as the criterion for trust region center selection after finding that it empirically lead to Pareto frontiers with good coverage, but many alternative heuristics could be used instead and might bear improved performance.

In Chapter 5, we use HV-KG to optimize the entire Pareto frontier, but a preference learning method using KG (e.g. by extending Astudillo and Frazier [2020]) could also be used in the partial information setting and may yield improved sample efficiency. Furthermore, alternative efficient forms of hypervolume computation

such as scalarization-based approaches [Golovin and Zhang, 2020] could prove useful for improving the wall time for computing HV-KG.

In Chapter 6, we propose to use probabilistic reparameterization for optimizing acquisition functions over discrete and mixed spaces, but it is expensive and in some cases, it may not be worth the additional computation. Further research into speeding up probabilistic reparameterization (potentially through more efficient methods for integration) would increase its applicability and enable using probabilistic reparameterization with more computationally intensive acquisition functions.

The contributions in the various sections of this thesis address important real world challenges for Bayesian optimization. However, we study these settings and address them individually. Even more challenging problems could feature attributes from more than one of these challenging scenarios. For example, optimizing optical displays for augmented and virtual reality involves high-dimensional multi-objective optimization as outlined in Chapter 4, but optical displays may also be subject to manufacturing tolerances during fabrication. Hence, the robustness of a solution might be an important consideration to avoid low yield, where many products do not meet the target specification. In addition, optimizing optical displays for augmented and virtual reality could benefit from exploiting observations at multiple fidelities such as fabrication (high fidelity) and simulation (lower fidelity). Lastly, there might be a mix of both discrete and continuous parameters. A future direction is the study how the proposed methods can be combined. Many of the proposed techniques introduce expectations in the acquisition functions such as over input noise, introduced discrete random variables, and partial information. Hence, one step is considering efficient computational techniques for computing these expectations in tandem. Another direction for future work is to consider robust multi-objective optimization with respect to environmental random variables. Obtaining an observation of the objective at a chosen fixed realization of the environmental random variables as in Cakmak et al. [2020] is another form of partial information. Lastly, the methods proposed in this thesis have limitations: MARS will be most useful for low-dimensional problems, HV-KG and probabilistic

reparameterization are more expensive than alternatives, and MORBO does not work quite as well on low-dimensional problems and may not be as efficient as other techniques [Eriksson and Jankowiak, 2021] with small sample budgets. Nevertheless, we have demonstrated that the contributions in this thesis improve Bayesian optimization in many *adverse scenarios*, and we hope that this work inspires future research in practical Bayesian optimization.

Bibliography

Robert J. Adler. An introduction to continuity, extrema, and related topics for general Gaussian processes. *Lecture Notes-Monograph Series*, 12:i–155, 1990. ISSN 07492170.

Taimoor Akhtar and Christine Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. *Journal of Global Optimization*, 64, 2015.

M.F. Ashby. Multi-objective optimization in material design and selection. *Acta Materialia*, 48(1):359–369, 2000. ISSN 1359-6454.

R. Astudillo and P. Frazier. Bayesian optimization of composite functions. *Forthcoming, in Proceedings of the 35th International Conference on Machine Learning*, 2019.

Raul Astudillo and Peter Frazier. Multi-attribute Bayesian optimization with interactive preference learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4496–4507. PMLR, 26–28 Aug 2020.

Raul Astudillo and Peter Frazier. Bayesian optimization of function networks. *Advances in Neural Information Processing Systems*, 34:14463–14475, 2021.

Raul Astudillo and Peter I. Frazier. Thinking inside the box: A tutorial on grey-box Bayesian optimization. In *Proceedings of the Winter Simulation Conference*, WSC '21. IEEE Press, 2022.

Gideon Avigad and Jürgen Branke. Embedded evolutionary multi-objective optimization for worst case robustness. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, page 617–624, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581309. doi: 10.1145/1389095.1389221.

Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011. doi: 10.1162/EVCO_a_00009.

Tao Bai, Yan-bin Kan, Jian-xia Chang, Qiang Huang, and Fi-John Chang. Fusing feasible search space into pso for multi-objective cascade reservoir optimization. *Appl. Soft Comput.*, 51(C), 2017.

Eytan Bakshy, Dean Eckles, and Michael S. Bernstein. Designing and deploying online field experiments. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, page 283–292. Association for Computing Machinery, 2014.

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*, 2020.

Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 462–471. PMLR, 10–15 Jul 2018.

Laurent Barcelo, John Kline, Gunther Walenta, and Ellis Gartner. Cement and carbon emissions. *Materials and structures*, 47(6):1055–1065, 2014.

R. Bartle. The elements of integration and Lebesgue measure. 1995.

Basel Committee on Banking Supervision. Fundamental review of the trading book. *Consultative Document*, 2012.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems 32*, 2019.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Multi-fidelity multi-objective Bayesian optimization: An output space entropy search approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10035–10043, Apr. 2020. doi: 10.1609/aaai.v34i06.6560.

Justin J. Beland and Prasanth B. Nair. Bayesian optimization under uncertainty. 2017.

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

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24*, pages 2546–2554, 2011.

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.

Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3), 2007.

Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization – a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33):3190–3218, 2007. ISSN 0045-7825.

J. Blank and K. Deb. pymoo: Multi-objective optimization in Python. *IEEE Access*, 8: 89497–89509, 2020.

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.

Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka, and Volkan Cevher. Adversarially robust optimization with Gaussian processes. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task Gaussian process prediction. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

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.

Eric Bradford, Artur M. Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *J. of Global Optimization*, 2018.

Karl Bringmann and Tobias Friedrich. Approximation quality of the hypervolume indicator. *Artificial Intelligence*, 195:265–290, 2013. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2012.09.005.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16:1190–1208, 1995.

Sait Cakmak, Raul Astudillo, Peter Frazier, and Enlu Zhou. Bayesian optimization of risk measures. In *Advances in Neural Information Processing Systems 33*, 2020.

Roberto Calandra and Jan Peters. Pareto front modeling for sensitivity analysis in multi-objective Bayesian optimization. In *NIPS 2014 Workshop on Bayesian Optimization in Academia and Industry*, 2014.

Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 533–557. PMLR, 25–28 Jun 2019.

Zefeng Chen, Yuren Zhou, Zhengxin Huang, and Xiaoyun Xia. Towards efficient multiobjective hyperparameter optimization: A multiobjective multi-fidelity Bayesian optimization and hyperband algorithm. In *Parallel Problem Solving from Nature – PPSN XVII: 17th International Conference*, page 160–174, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-14713-5. doi: 10.1007/978-3-031-14714-2_12.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 844–853. JMLR.org, 2017.

Tinkle Chugh. Scalarizing functions in Bayesian multiobjective optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020. doi: 10.1109/CEC48606.2020.9185706.

Ivo Couckuyt, Dirk Deschrijver, and Tom Dhaene. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *J. of Global Optimization*, 60(3):575–594, November 2014.

Areski Cousin and Elena Di Bernardino. On multivariate extensions of conditional-tail-expectation. *Insurance: Mathematics and Economics*, 55:272–282, 2014.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9851–9864. Curran Associates, Inc., 2020.

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.

Samuel Daulton, Sait Cakmak, Maximilian Balandat, Michael A. Osborne, Enlu Zhou, and Eytan Bakshy. Robust multi-objective Bayesian optimization under input noise. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4831–4866. PMLR, 17–23 Jul 2022a.

Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective Bayesian optimization over high-dimensional search spaces. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 507–517. PMLR, 01–05 Aug 2022b.

Samuel Daulton, Xingchen Wan, David Eriksson, Maximilian Balandat, Michael A Osborne, and Eytan Bakshy. Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 12760–12774. Curran Associates, Inc., 2022c.

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

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.

Nando de Freitas, Alex Smola, and Masrour Zoghi. Exponential regret bounds for Gaussian process bandits with deterministic observations. In *ICML*, 2012.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197, 2002.

Kalyan Deb, L. Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. volume 1, 2002.

Kalyanmoy Deb and Himanshu Gupta. Searching for robust Pareto-optimal solutions in multi-objective optimization. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization*, pages 150–164, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

Kalyanmoy Deb and J Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006.

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.

Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2632–2643. PMLR, 18–24 Jul 2021a.

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, 2021b.

Bach Do, Makoto Ohsaki, and Makoto Yamakawa. Bayesian optimization for robust design of steel frames with joint and individual probabilistic constraints. *Engineering Structures*, 245:112859, 2021. ISSN 0141-0296.

Ioannis Doltsinis and Zhan Kang. Robust design of structures using optimization methods. *Computer Methods in Applied Mechanics and Engineering*, 193(23): 2221–2237, 2004. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2003.12.055.

Xuanyi Dong and Yi Yang. NAS-Bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations (ICLR)*, 2020.

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.

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

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.

G. Emch and A. Parkinson. Robust Optimal Design for Worst-Case Tolerances. *Journal of Mechanical Design*, 116(4):1019–1025, 12 1994. ISSN 1050-0472. doi: 10.1115/1.2919482.

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.

M. T. M. Emmerich, A. H. Deutz, and J. W. Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011.

Michael T. M. Emmerich and Carlos M. Fonseca. Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results. In *Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg, 2011.

David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*. PMLR, 2021.

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

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.

David Eriksson, Pierce I-Jen Chuang, Samuel Daulton, Peng Xia, Akshat Shrivastava, Arun Babu, Shicong Zhao, Ahmed Aly, Ganesh Venkatesh, and Maximilian Balandat. Latency-aware neural architecture search with multi-objective Bayesian optimization. In *ICML Workshop on AutoML*, 2021.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1436–1445, 2018.

Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Peter I Frazier, Warren B Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5): 2410–2439, 2008.

Lukas Fröhlich, Edgar Klenske, Julia Vinogradska, Christian Daniel, and Melanie Zeilinger. Noisy-input entropy search for efficient robust Bayesian optimization. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2262–2272. PMLR, 26–28 Aug 2020.

Jacob Gardner, Matt Kusner, Zhixiang, Kilian Weinberger, and John Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 937–945, Beijing, China, 22–24 Jun 2014. PMLR.

Jacob R. Gardner, Chuan Guo, Kilian Q. Weinberger, Roman Garnett, and Roger B. Grosse. Discovering and exploiting additive structure for Bayesian optimization. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, 2017.

Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

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.

Eduardo C. Garrido-Merchán, Daniel Fernández-Sánchez, and Daniel Hernández-Lobato. Parallel predictive entropy search for multi-objective Bayesian optimization with constraints applied to the tuning of machine learning algorithms. *Expert Systems with Applications*, 215:119328, 2023. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.119328.

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

Daniel Golovin and Qiuyi Zhang. Random hypervolume scalarizations for provable multi-objective black box optimization, 2020.

Abhijith M. Gopakumar, Prasanna V. Balachandran, Dezhen Xue, James E. Gubernatis, and Turab Lookman. Multi-objective optimization for materials discovery via adaptive design. *Scientific Reports*, 8(1), 2018.

Ryan-Rhys Griffiths and Jose Miguel Hernandez-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.*, 11, 2020.

Julia Guerrero-Viu, Sven Hauns, Sergio Izquierdo, Guilherme Miotto, Simon Schrodi, Andre Biedenkapp, Thomas Elsken, Difan Deng, Marius Lindauer, and Frank Hutter. Bag of baselines for multi-objective joint neural architecture search and hyperparameter optimization. In *ICML 2021 Workshop on AutoML*, 2021.

Himanshu Gupta and Kalyanmoy Deb. Handling constraints in robust multi-objective optimization. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 25–32 Vol.1, 2005. doi: 10.1109/CEC.2005.1554663.

Nikolaus Hansen. *The CMA Evolution Strategy: A Comparing Review*, volume 192, pages 75–102. 06 2007. doi: 10.1007/3-540-32494-1_4.

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.

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.

Youwei He, Jinju Sun, Peng Song, and Xuesong Wang. Variable-fidelity hypervolume-based expected improvement criteria for multi-objective efficient global optimization of expensive functions. *Eng. with Comput.*, 38(4):3663–3689, aug 2022. ISSN 0177-0667. doi: 10.1007/s00366-021-01404-9.

Zhenan He, Gary G. Yen, and Zhang Yi. Robust multiobjective optimization via evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 23(2): 316–330, 2019. doi: 10.1109/TEVC.2018.2859638.

Philipp Hennig and Christian J. Schuler. Entropy search for information-efficient global optimization. *J. Mach. Learn. Res.*, 13(null):1809–1837, jun 2012. ISSN 1532-4435.

Daniel Hernandez-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective Bayesian optimization. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

José Miguel Hernández-Lobato, Matthew W. Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 918–926, Cambridge, MA, USA, 2014. MIT Press.

José Miguel Hernández-Lobato, Michael A. Gelbart, Ryan P. Adams, Matthew W. Hoffman, and Zoubin Ghahramani. A general framework for constrained Bayesian optimization using information-based search. *Journal of Machine Learning Research*, 17(160):1–53, 2016.

Tito Homem-de-Mello. On rates of convergence for stochastic optimization problems under non-independent and identically distributed sampling. *SIAM Journal on Optimization*, 19(2):524–551, 2008.

Christopher A. Hone, Nicholas Holmes, Geoffrey R. Akien, Richard A. Bourne, and Frans L. Muller. Rapid multistep kinetic model generation from transient flow data. *React. Chem. Eng.*, 2:103–108, 2017. doi: 10.1039/C6RE00109B. URL `http://dx.doi.org/10.1039/C6RE00109B`.

L. Jeff Hong. Estimating quantile sensitivities. *Operations Research*, 57(1):118–130, 2009. doi: 10.1287/opre.1080.0531.

L Jeff Hong, Zhaolin Hu, and Guangwu Liu. Monte Carlo methods for value-at-risk and conditional value-at-risk: a review. *ACM Transactions on Modeling and Computer Simulation*, 24(4):1–37, 2014.

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.

Jonas Ide and Elisabeth Köbis. Concepts of efficiency for uncertain multi-objective optimization problems based on set order relations. *Mathematical Methods of Operations Research*, 80, 08 2014. doi: 10.1007/s00186-014-0471-z.

Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. Ai benchmark: All about deep learning on smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3617–3635. IEEE, 2019.

F. Irshad, S. Karsch, and A. Döpp. Multi-objective and multi-fidelity Bayesian optimization of laser-plasma acceleration. *Phys. Rev. Res.*, 5:013063, Jan 2023a. doi: 10.1103/PhysRevResearch.5.013063.

Faran Irshad, Stefan Karsch, and Andreas Döpp. Expected hypervolume improvement for simultaneous multi-objective and multi-fidelity optimization, 2021.

Faran Irshad, Stefan Karsch, and Andreas Doepp. Reference dataset of multi-objective and multi- fidelity optimization in laser-plasma acceleration, January 2023b.

Hisao Ishibuchi, Naoya Akedo, and Yusuke Nojima. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11, page 649–656, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450305570. doi: 10.1145/2001576.2001666.

Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evol. Comput.*, 26(3), 2018.

Shogo Iwazaki, Yu Inatsu, and Ichiro Takeuchi. Mean-variance analysis in Bayesian optimization under uncertainty. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 973–981. PMLR, 13–15 Apr 2021.

Sergio Izquierdo, Julia Guerrero-Viu, Sven Hauns, Guilherme Miotto, Simon Schrodi, André Biedenkapp, Thomas Elsken, Difan Deng, Marius Lindauer, and Frank Hutter. Bag of baselines for multi-objective joint neural architecture search and hyperparameter optimization. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021.

Moksh Jain, Sharath Chandra Raparthy, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets, 2022.

Vijay Janapa Reddi, David Kanter, Peter Mattson, Jared Duke, Thai Nguyen, Ramesh Chukka, Ken Shiring, Koan-Sin Tan, Mark Charlebois, William Chou, et al. Mlperf mobile inference benchmark: An industry-standard open-source machine learning benchmark for on-device ai. *Proceedings of Machine Learning and Systems*, 4:352–369, 2022.

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

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, 1998.

Miettinen Kaisa. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Boston, USA, 1999.

Kirthevasan Kandasamy, Jeff Schneider, and Barnab'as P'oczos. High dimensional Bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015a.

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 2015b. PMLR.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, 2013.

Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally robust Bayesian optimization. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2174–2184. PMLR, 26–28 Aug 2020.

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.

J. Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

Takehisa Kohira, Hiromasa Kemmotsu, Oyama Akira, and Tomoaki Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018.

Mina Konakovic Lukovic, Yunsheng Tian, and Wojciech Matusik. Diversity-guided multi-objective Bayesian optimization with batch evaluations. *Advances in Neural Information Processing Systems*, 33, 2020.

Shunya Kusakawa, Shion Takeno, Yu Inatsu, Kentaro Kutsukake, Shogo Iwazaki, Takashi Nakano, Toru Ujihara, Masayuki Karasuyama, and Ichiro Takeuchi. Bayesian Optimization for Cascade-Type Multistage Processes. *Neural Computation*, 34(12): 2408–2431, 11 2022. ISSN 0899-7667. doi: 10.1162/neco_a_01550.

Renaud Lacour, Kathrin Klamroth, and Carlos M. Fonseca. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, 79, 2017.

Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1546–1558. Curran Associates, Inc., 2020.

Benjamin Letham and Eytan Bakshy. Bayesian optimization for policy search via online-offline experimentation. *Journal of Machine Learning Research*, 20(145):1–30, 2019.

Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2), 06 2019.

Chaojian Li, Zhongzhi Yu, Yonggan Fu, Yongan Zhang, Yang Zhao, Haoran You, Qixuan Yu, Yue Wang, Cong Hao, and Yingyan Lin. {HW}-{nas}-bench: Hardware-aware neural architecture search benchmark. In *International Conference on Learning Representations*, 2021.

Xin Li and Chao-Ping Chen. Inequalities for the gamma function. *Journal of Inequalities in Pure and Applied Mathematics*, 8, 2007.

Yucen Lily Li, Tim G. J. Rudner, and Andrew Gordon Wilson. A study of Bayesian neural network surrogates for Bayesian optimization, 2023.

Zhuangzhi Li and Zukui Li. Optimal robust optimization approximation for chance constrained optimization problem. *Computers & Chemical Engineering*, 74:89–99, 2015.

Qiaohao Liang and Lipeng Lai. Scalable Bayesian optimization accelerates process optimization of penicillin production. In *NeurIPS 2021 AI for Science Workshop*, 2021.

Xingtao Liao, Qing Li, Xujing Yang, Weigang Zhang, and Wei Li. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization*, 35, 2008.

Sulin Liu, Qing Feng, David Eriksson, Benjamin Letham, and Eytan Bakshy. Sparse Bayesian optimization. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, AISTATS, page *In press*, 2023.

Edgar Manoatl Lopez, Luis Miguel Antonio, and Carlos A. Coello Coello. A gpu-based algorithm for a faster hypervolume contribution computation. In António Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization*, pages 80–94. Springer International Publishing, 2015.

Ilya Loshchilov, Marc Schoenauer, and Michèle Sebag. Not all parents are equal for MO-CMA-ES. In *Evolutionary Multi-Criterion Optimization*, 2011.

Mina Konakovic Lukovic, Yunsheng Tian, and Wojciech Matusik. Diversity-Guided Multi-Objective Bayesian Optimization With Batch Evaluations. In *Advances in Neural Information Processing Systems 33*, 2020.

Zhongwei Ma and Yong Wang. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Transactions on Evolutionary Computation*, 23(6), 2019.

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.

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.

Dinah Elena Majewski, Marco Wirtz, Matthias Lampe, and André Bardow. Robust multi-objective optimization for sustainable design of distributed energy supply systems. *Computers & Chemical Engineering*, 102:26–39, 2017. ISSN 0098-1354. Sustainability & Energy Systems.

Gustavo Malkomes, Bolong Cheng, Eric H Lee, and Mike Mccourt. Beyond the Pareto efficient frontier: Constraint active search for multiobjective experimental design. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7423–7434. PMLR, 18–24 Jul 2021.

R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6), 2004.

Julien Marzat, Eric Walter, and Hélène Piet-Lahanier. Worst-case global optimization of black-box functions through kriging and relaxation. *Journal of Global Optimization*, 55:707–727, 04 2013. doi: 10.1007/s10898-012-9899-y.

Alexandre Mathern, Olof Steinholtz, Anders Sjöberg, Magnus Önnheim, Kristine Ek, Rasmus Rempling, Emil Gustavsson, and Mats Jirstrand. Multi-objective constrained Bayesian optimization for structural design. *Structural and Multidisciplinary Optimization*, 63, 2021.

Merve Meraklı and Simge Küçükyavuz. Vector-valued multivariate conditional value-at-risk. *Operations Research Letters*, 46(3):300–305, 2018. ISSN 0167-6377. doi: https://doi.org/10.1016/j.orl.2018.02.006.

Richard M Meyer. *Essential mathematics for applied fields*. Springer Science & Business Media, 2012.

Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601, 2002.

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.

Séverine Thérèse F.C. Mortier, Pieter-Jan Van Bockstal, Jos Corver, Ingmar Nopens, Krist V. Gernaey, and Thomas De Beer. Uncertainty analysis as essential step in the establishment of the dynamic design space of primary drying during freeze-drying. *European Journal of Pharmaceutics and Biopharmaceutics*, 103:71–83, 2016. ISSN 0939-6411.

Henry Moss, David Leslie, Daniel Beck, Javier González, and Paul Rayson. Boss: Bayesian optimization over string spaces. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15476–15486. Curran Associates, Inc., 2020.

Fatemah Mukadum, Quan Nguyen, Daniel M. Adrion, Gabriel Appleby, Rui Chen, Haley Dang, Remco Chang, Roman Garnett, and Steven A. Lopez. Efficient discovery of visible light-activated azoarene photoswitches with long half-lives using active search. *Journal of Chemical Information and Modeling*, 61(11):5524–5534, 2021. doi: 10.1021/acs.jcim.1c00954.

Alex Munteanu, Amin Nayebi, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Mojmir Mutny and Andreas Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature fourier features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Dang Nguyen, Sunil Gupta, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Bayesian optimization for categorical and category-specific continuous inputs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5256–5263, Apr. 2020a. doi: 10.1609/aaai.v34i04.5971.

Quoc Phong Nguyen, Zhongxiang Dai, Bryan Kian Hsiang Low, and Patrick Jaillet. Optimizing conditional value-at-risk of black-box functions. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021a.

Quoc Phong Nguyen, Zhongxiang Dai, Bryan Kian Hsiang Low, and Patrick Jaillet. Value-at-risk optimization with Gaussian processes. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8063–8072. PMLR, 18–24 Jul 2021b.

Vu Nguyen, Sebastian Schulze, and Michael Osborne. Bayesian optimization for iterative learning. *Advances in Neural Information Processing Systems*, 33:9361–9371, 2020b.

ChangYong Oh, Efstratios Gavves, and Max Welling. BOCK: Bayesian optimization with cylindrical kernels. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 2018.

Changyong Oh, Jakub M. Tomczak, Efstratios Gavves, and Max Welling. *Combinatorial Bayesian Optimization Using the Graph Cartesian Product*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Rafael Oliveira, Lionel Ott, and Fabio Ramos. Bayesian optimisation under uncertain inputs. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1177–1184. PMLR, 16–18 Apr 2019.

M. Osborne and University of Oxford. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, Oxford University New College, 2010.

Art B Owen. Scrambling sobol'and niederreiter–xing points. *Journal of complexity*, 14(4): 466–489, 1998.

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

Akira Oyama, Takehisa Kohira, Hiromasa Kemmotsu, Tomoaki Tatsukawa, and Takeshi Watanabe. Simultaneous structure design optimization of multiple car models using the K computer. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017.

C.J. Paciorek. *Nonstationary Gaussian Processes for Regression and Spatial Modelling*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2003.

Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A flexible framework for multi-objective Bayesian optimization using random scalarizations. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115, 2020.

Ji Won Park, Samuel Don Stanton, Saeed Saremi, Andrew Martin Watkins, Henri Dwyer, Vladimir Gligorijevic, Richard Bonneau, Stephen Ra, and Kyunghyun Cho. PropertyDAG: Multi-objective Bayesian optimization of partially ordered, mixed-variable properties for biological sequence design. In *NeurIPS 2022 AI for Science: Progress and Promises*, 2022.

Michael Parsons and Randall Scott. Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods. *Journal of Ship Research*, 48:61–76, 03 2004. doi: 10.5957/jsr.2004.48.1.61.

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.

Victor Picheny, Henry Moss, Léonard Torossian, and Nicolas Durrande. Bayesian quantile and expectile optimisation. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 1623–1633. PMLR, 01–05 Aug 2022.

Geoff Pleiss, Martin Jankowiak, David Eriksson, Anil Damle, and Jacob Gardner. Fast matrix square roots with applications to Gaussian processes and Bayesian optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Matthias Poloczek, Jialei Wang, and Peter Frazier. Multi-information source optimization. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, and Markus Vincze. Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection. In Günter Rudolph, Thomas Jansen, Nicola Beume, Simon Lucas, and Carlo Poloni, editors, *Parallel Problem Solving from Nature – PPSN X*, pages 784–794, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

Andràs Prèkopa. Multivariate value at risk and related topics. *Annals of Operations Research*, 193:49–69, 2012.

Saba Q. Yahyaa, Madalina M. Drugan, and Bernard Manderick. Knowledge gradient for multi-objective multi-armed bandit algorithms. In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence - Volume 1*, ICAART 2014, page 74–83, Setubal, PRT, 2014. SCITEPRESS - Science and Technology Publications, Lda. ISBN 9789897580154. doi: 10.5220/0004796600740083.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

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

Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

Tapabrata Ray and K.M. Liew. A swarm metaphor for multiobjective design optimization. *Engineering Optimization*, 34(2):141–153, 2002. doi: 10.1080/03052150210915.

Rommel G. Regis and Christine A. Shoemaker. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization*, 45(5), 2013.

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

Justo José Roberts, Agnelo Marotta Cassula, José Luz Silveira, Edson da Costa Bortoni, and Andrés Z. Mendiburu. Robust multi-objective optimization of a renewable based hybrid power system. *Applied Energy*, 223:52–68, 2018. ISSN 0306-2619.

R.Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7), 2002. ISSN 0378-4266. doi: 10.1016/S0378-4266(02)00271-6.

Binxin Ru, Mark McLeod, Diego Granziol, and Michael A. Osborne. Fast information-theoretic Bayesian optimisation. In *International Conference on Machine Learning (ICML)*, 2018.

Binxin Ru, Ahsan Alvi, Vu Nguyen, Michael A. Osborne, and Stephen Roberts. Bayesian optimisation over multiple continuous and categorical inputs. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8276–8285. PMLR, 13–18 Jul 2020.

Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Math. Oper. Res.*, 39(4):1221–1243, nov 2014. ISSN 0364-765X. doi: 10.1287/moor.2014.0650.

Soumya 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. *Wireless Personal Communications*, 128:1–16, 09 2022. doi: 10.1007/s11277-022-09969-4.

Sergey Sarykalin, Gaia Serraino, and Stan Uryasev. Value- at-risk vs conditional value-at-risk in risk management and optimization. 09 2008. ISBN 978-1-877640-23-0. doi: 10.1287/educ.1080.0052.

Warren Scott, Peter Frazier, and Warren Powell. The correlated knowledge gradient for simulation optimization of continuous parameters using Gaussian process regression. *SIAM Journal on Optimization*, 21:996–1026, 07 2011. doi: 10.1137/100801275.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Robert J. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley Series in Probability and Statistics. Wiley-Blackwell, 2008. ISBN 9780470316481. doi: 10.1002/9780470316481.

Pier Giuseppe Sessa, Ilija Bogunovic, Maryam Kamgarpour, and Andreas Krause. Mixed strategies for robust optimization of unknown objectives. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2970–2980. PMLR, 26–28 Aug 2020.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104, 2016.

Ke Shang, Weiyu Chen, Weiduo Liao, and Hisao Ishibuchi. Hv-net: Hypervolume approximation based on deepsets. *IEEE Transactions on Evolutionary Computation*, pages 1–1, 2022. doi: 10.1109/TEVC.2022.3181306.

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.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010.

Shinya Suzuki, Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Multi-objective Bayesian optimization using Pareto-frontier entropy. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, 2020.

G. Taguchi. *Einführung in Quality engineering: Minimierung von Verlusten durch Prozeßbeherrschung*. American Supplier Inst., 1989. ISBN 9789283310846.

Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9334–9345. PMLR, 13–18 Jul 2020.

Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89, 2020.

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

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Jose Antonio Garrido Torres, Sii Hong Lau, Pranay Anchuri, Jason M. Stevens, Jose E. Tabora, Jun Li, Alina Borovika, Ryan P. Adams, and Abigail G. Doyle. A multi-objective active learning platform and web app for reaction optimization. *Journal of the American Chemical Society*, 144(43):19999–20007, 2022. doi: 10.1021/jacs.2c08592. PMID: 36260788.

Saul Toscano-Palmerin and Peter I. Frazier. Bayesian optimization with expensive integrands. *SIAM Journal on Optimization*, 32(2):417–444, 2022. doi: 10.1137/19M1303125.

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.

Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective Bayesian optimization. In *Advances in Neural Information Processing Systems 35*, 2022.

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, 2021.

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.

Samee ur Rehman, Matthijs Langelaar, and Fred van Keulen. Efficient kriging-based robust optimization of unconstrained problems. *Journal of Computational Science*, 5 (6):872–881, 2014. ISSN 1877-7503.

Xingchen Wan, Vu Nguyen, Huong Ha, Binxin Ru, Cong Lu, and Michael A. Osborne. Think global and act local: Bayesian optimisation over high-dimensional categorical and mixed search spaces. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10663–10674. PMLR, 18–24 Jul 2021.

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.

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.

Jialei Wang, Scott Clark, Eric Liu, and Peter Frazier. Parallel Bayesian global optimization of expensive functions. *Operations Research*, 68, 02 2016a. doi: 10.1287/opre.2019.1966.

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, 2017.

Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning (ICML)*, 2017.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, volume 84, 2018.

Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Freitas. Bayesian optimization in a billion dimensions via random embeddings. *J. Artif. Int. Res.*, 55(1), 2016b.

Colin White, Willie Neiswanger, and Yash Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10293–10301, May 2021. doi: 10.1609/aaai.v35i12.17233.

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

James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for Bayesian optimization. In *Advances in Neural Information Processing Systems 31*, pages 9905–9916. 2018.

James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*, 2020.

Jian Wu and Peter Frazier. The parallel knowledge gradient method for batch Bayesian optimization. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Jian Wu, Matthias Poloczek, Andrew G Wilson, and Peter Frazier. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, pages 5267–5278, 2017.

Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical multi-fidelity Bayesian optimization for hyperparameter tuning. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 788–798. PMLR, 22–25 Jul 2020a.

Tony C. Wu, Daniel Flam-Shepherd, and Alán Aspuru-Guzik. Bayesian variational optimization for combinatorial spaces, 2020b.

Xiangzhong Xie and René Schenkendorf. Robust process design in pharmaceutical manufacturing under batch-to-batch variation. *Processes*, 7(8), 2019. ISSN 2227-9717. doi: 10.3390/pr7080509.

Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. *Swarm and Evolutionary Computation*, 44:945 – 956, 2019. ISSN 2210-6502.

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.

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

Ya-xiang Yuan. A review of trust region algorithms for optimization. *ICM99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics*, 1999.

Qingfu Zhang, Wudong Liu, Edward Tsang, and Botond Virginas. Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Transactions on Evolutionary Computation*, 14(3), 2010.

Richard Zhang and Daniel Golovin. Random hypervolume scalarizations for provable multi-objective black box optimization. In *International Conference on Machine Learning*, 2020.

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.

Yiyang Zhao, Linnan Wang, Kevin Yang, Tianjun Zhang, Tian Guo, and Yuandong Tian. Multi-objective optimization by learning space partition. In *International Conference on Learning Representations*, 2022.

Aimin Zhou, Qingfu Zhang, and Guixu Zhang. A multiobjective evolutionary algorithm based on decomposition and probability model. In *2012 IEEE Congress on Evolutionary Computation*, 2012.

Qi Zhou, Ping Jiang, Xiang Huang, Feng Zhang, and Taotao Zhou. A multi-objective robust optimization approach based on Gaussian process model. *Structural and Multidisciplinary Optimization*, 57, 01 2018. doi: 10.1007/s00158-017-1746-9.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, 1997. ISSN 0098-3500. doi: 10.1145/279232.279236.

E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, jun 2000. ISSN 1063-6560. doi: 10.1162/106365600568202.

Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization*, EMO'07, page 862–876, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 9783540709275.

Eckart Zitzler, Joshua Knowles, and Lothar Thiele. *Quality Assessment of Pareto Set Approximations*, page 373–404. Springer-Verlag, Berlin, Heidelberg, 2008. ISBN 9783540889076.