Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization

Samuel Daulton
Facebook
sdaulton@fb.com

Maximilian Balandat
Facebook
balandat@fb.com

Eytan Bakshy
Facebook
ebakshy@fb.com

Abstract

In many real-world scenarios, decision makers seek to efficiently optimize multiple competing objectives in a sample-efficient fashion. Multi-objective Bayesian optimization (BO) is a common approach, but many of the best-performing acquisition functions do not have known analytic gradients and suffer from high computational overhead. We leverage recent advances in programming models and hardware acceleration for multi-objective BO using Expected Hypervolume Improvement (EHVI)—an algorithm notorious for its high computational complexity. We derive a novel formulation of q-Expected Hypervolume Improvement (qEHVI), an acquisition function that extends EHVI to the parallel, constrained evaluation setting. qEHVI is an exact computation of the joint EHVI of q new candidate points (up to Monte-Carlo (MC) integration error). Whereas previous EHVI formulations rely on gradient-free acquisition optimization or approximated gradients, we compute exact gradients of the MC estimator via auto-differentiation, thereby enabling efficient and effective optimization using first-order and quasi-second-order methods. Our empirical evaluation demonstrates that qEHVI is computationally tractable in many practical scenarios and outperforms state-of-the-art multi-objective BO algorithms at a fraction of their wall time.

1 Introduction

The problem of optimizing multiple competing objectives is ubiquitous in scientific and engineering applications. For example in automobile design, an automaker will want to maximize vehicle durability and occupant safety, while using lighter materials that afford increased fuel efficiency and lower manufacturing cost [44, 72]. Evaluating the crash safety of an automobile design experimentally is expensive due to both the manufacturing time and the destruction of a vehicle. In such a scenario, sample efficiency is paramount. For a different example, video streaming web services commonly use adaptive control policies to determine the bitrate as the stream progresses in real time [47]. A decision maker may wish to optimize the control policy to maximize the quality of the video stream, while minimizing the stall time. Policy evaluation typically requires using the suggested policy on segments of live traffic, which is subject to opportunity costs. If long evaluation times are the limiting factor, multiple designs may be evaluated in parallel to significantly decrease end-to-end optimization time. For example, an automaker could manufacture multiple vehicle designs in parallel or a web service could deploy several control policies to different segments of traffic at the same time.

1.1 Background

Multi-Objective Optimization: In this work, we address the problem of optimizing a *vector-valued objective* $\boldsymbol{f}(\boldsymbol{x}) : R^d \to \mathbb{R}^M$ with $\boldsymbol{f}(\boldsymbol{x}) = \big(f^{(1)}(\boldsymbol{x}), ..., f^{(M)}(\boldsymbol{x})\big)$ over a bounded set $\mathcal{X} \subset \mathbb{R}^d$. We consider the scenario in which the $f^{(i)}$ are expensive-to-evaluate black-box functions with

no known analytical expression, and no observed gradients. *Multi-objective* (MO) optimization problems typically do not have a single best solution; rather, the goal is to identify the set of *Pareto optimal* solutions such that any improvement in one objective means deteriorating another. Without loss of generality, we assume the goal is to maximize all objectives. We say a solution $\boldsymbol{f}(\boldsymbol{x})$ *Pareto dominates* another solution $\boldsymbol{f}(\boldsymbol{x}')$ if $f^{(m)}(\boldsymbol{x}) \geq f^{(m)}(\boldsymbol{x}') \; \forall \; m = 1, \ldots, M$ and there exists $m' \in \{1, \ldots, M\}$ such that $f^{(m')}(\boldsymbol{x}) > f^{(m')}(\boldsymbol{x}')$. We write $\boldsymbol{f}(\boldsymbol{x}) \succ \boldsymbol{f}(\boldsymbol{x}')$. Let $\mathcal{P}^* = \{\boldsymbol{f}(\boldsymbol{x}) \; s.t. \; \nexists \, \boldsymbol{x}' \in \mathcal{X} \; : \; \boldsymbol{f}(\boldsymbol{x}') \succ \boldsymbol{f}(\boldsymbol{x})\}$ and $\mathcal{X}^* = \{\boldsymbol{x} \in \mathcal{X} \; s.t. \; \boldsymbol{f}(\boldsymbol{x}) \in \mathcal{P}^*\}$ denote the set of Pareto optimal solutions and Pareto optimal inputs, respectively. Provided with the Pareto set, decision-makers can select a solution with an objective trade-off according to their preferences.

A common approach for solving MO problems is to use evolutionary algorithms (e.g. NSGA-II), which are robust multi-objective optimizers, but require a large number of function evaluations [14]. Bayesian optimization (BO) offers a far more sample-efficient alternative [57].

Bayesian Optimization: BO [38] is an established method for optimizing expensive-to-evaluate black-box functions. BO relies on a probabilistic *surrogate model*, typically a Gaussian Process (GP) [55], to provide a posterior distribution $\mathbb{P}(\boldsymbol{f}|\mathcal{D})$ over the true function values $\boldsymbol{f}$ given the observed data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$. An *acquisition function* $\alpha : \mathcal{X}_{\text{cand}} \mapsto \mathbb{R}$ employs the surrogate model to assign a utility value to a set of candidates $\mathcal{X}_{\text{cand}} = \{\boldsymbol{x}_i\}_{i=1}^q$ to be evaluated on the true function. While the true $\boldsymbol{f}$ may be expensive-to-evaluate, the surrogate-based acquisition function is not, and can thus be efficiently optimized to yield a set of candidates $\mathcal{X}_{\text{cand}}$ to be evaluated on $\boldsymbol{f}$. If gradients of $\alpha(\mathcal{X}_{\text{cand}})$ are available, gradient-based methods can be utilized. If not, gradients are either approximated (e.g. with finite differences) or gradient-free methods (e.g. DIRECT [37] or CMA-ES [32]) are used.

1.2 Limitations of current approaches

In the single-objective (SO) setting, a large body of work focuses on practical extensions to BO for supporting *parallel* evaluation and outcome constraints [49, 30, 66, 25, 43]. Less attention has been given to such extensions in the MO setting. Moreover, the existing constrained and parallel MO BO options have limitations: 1) many rely on scalarizations to transform the MO problem into a SO one [40]; 2) many acquisition functions are computationally expensive to compute [52, 21, 6, 71]; 3) few have known analytical gradients or are differentiable [19, 62, 33]; 4) many rely on heuristics to extend sequential algorithms to the parallel setting [27, 62].

A natural acquisition function for MO BO is Expected Hypervolume Improvement (EHVI). Maximizing the hypervolume (HV) has been shown to produce Pareto fronts with excellent coverage [73, 12, 69]. However, there has been little work on EHVI in the parallel setting, and the work that has been done resorts to approximate methods [71, 28, 62]. A vast body of literature has focused on efficient EHVI computation [34, 20, 67], but the time complexity for computing EHVI is exponential in the number of objectives—in part due the hypervolume indicator itself incurring a time complexity that scales super-polynomially with the number of objectives [68]. Our core insight is that by exploiting advances in auto-differentiation and highly parallelized hardware [51], we can make EHVI computations fast and practical.

1.3 Contributions

In this work, we derive a novel formulation of the parallel q-Expected Hypervolume Improvement acquisition function (qEHVI) that is exact up to Monte-Carlo (MC) integration error. We compute the *exact* gradient of the MC estimator of qEHVI using auto-differentiation, which allows us to employ efficient and effective gradient-based optimization methods. Rather than using first-order gradient methods, we instead leverage the sample average approximation (SAA) approach from [5] to use higher-order deterministic optimization methods, and we prove theoretical convergence guarantees under the SAA approach. Our formulation of qEHVI is embarrassingly parallel, and despite its computational cost would achieve constant time complexity given infinite processing cores. We demonstrate that, using modern GPU hardware and computing exact gradients, optimizing qEHVI is faster than existing state-of-the art methods in many practical scenarios. Moreover, we extend qEHVI to support auxiliary outcome constraints, making it practical in many real-world scenarios. Lastly, we demonstrate how modern auto-differentiation can be used to compute exact gradients of analytic EHVI, which has never been done before for $M > 2$ objectives. Our empirical evaluation

shows that qEHVI outperforms state-of-the-art multi-objective BO algorithms while using only a fraction of their wall time.

2 Related Work

Yang et al. [69] is the only previous work to consider exact gradients of EHVI, but the authors only derive an analytical gradient for the unconstrained two-objective, sequential optimization setting. All other works either do not optimize EHVI (e.g. they use it for pre-screening candidates [18]), optimize it with gradient-free methods [68], or using approximate gradients [62]. In contrast, we use exact gradients and demonstrate that optimizing EHVI using this gradient information is far more efficient.

There are many alternatives to EHVI for MO BO. For example, ParEGO [40] and TS-TCH [50] randomly scalarize the objectives and use Expected Improvement [38] and Thompson Sampling [61], respectively. SMS-EGO [53] uses HV in a UCB-based acquisition function and is more scalable than EHVI [54]. ParEGO and SMS-EGO have only been considered for the $q = 1$, unconstrained setting. Predictive entropy search for MO BO (PESMO) [33] has been shown to be another competitive alternative and has been extended to handle constraints [26] and parallel evaluations [27]. MO max-value entropy search (MO-MES) has been shown to achieve superior optimization performance and faster wall times than PESMO, but is limited to $q = 1$.

Wilson et al. [65] empirically and theoretically show that *sequential greedy* selection of q candidates achieves performance comparable to jointly optimizing q candidates for many acquisition functions (including [63, 66]). The sequential greedy approach integrates over the posterior of the unobserved outcomes corresponding to the previously selected candidates in the q-batch. Sequential greedy optimization often yields better empirical results because the optimization problem has a lower dimension: d in each step, rather than qd in the joint problem. Most prior works in the MO setting use a sequential greedy approximation or heuristics [62, 71, 28, 10], but impute the unobserved outcomes with the posterior mean rather than integrating over the posterior [30]. For many joint acquisition functions involving expectations, this shortcut sacrifices the theoretical error bound on the sequential greedy approximation because the exact joint acquisition function over $\boldsymbol{x}_1, ..., \boldsymbol{x}_i$, $1 \le i \le q$ requires integration over the joint posterior $\mathbb{P}(\boldsymbol{f}(\boldsymbol{x}_1), ..., \boldsymbol{f}(\boldsymbol{x}_q) | \mathcal{D})$ and is not computed for $i > 1$.

Garrido-Merchán and Hernández-Lobato [27] and Wada and Hino [62] jointly optimize the q candidates and, noting the difficulty of the optimization, both papers focus on deriving gradients to aid in the optimization. Wada and Hino [62] defined the qEHVI acquisition function, but after finding it challenging to optimize q candidates jointly (without exact gradients), the authors propose optimizing an alternative acquisition function instead of exact qEHVI. In contrast, our novel qEHVI formulation allows for gradient-based parallel and sequential greedy optimization, with proper integration over the posterior for the latter.

Feliot et al. [22] and Abdolshah et al. [1] proposed extensions of EHVI to the constrained $q = 1$ setting, but neither considers the batch setting and both rely on gradient-free optimization.

3 Differentiable q-Expected Hypervolume Improvement

In this section, we review HVI and EHVI computation by means of box decompositions, and explain our novel formulation for the parallel setting.

Definition 1. *Given a reference point $\boldsymbol{r} \in \mathbb{R}^M$, the hypervolume indicator (HV) of a finite approximate Pareto set $\mathcal{P}$ is the M-dimensional Lebesgue measure λ_M of the space dominated by $\mathcal{P}$ and bounded from below by $\boldsymbol{r}$: $\mathrm{HV}(\mathcal{P}, \boldsymbol{r}) = \lambda_M \big(\bigcup_{i=1}^{|\mathcal{P}|} [\boldsymbol{r}, \boldsymbol{y}_i] \big)$, where $[\boldsymbol{r}, \boldsymbol{y}_i]$ denotes the hyper-rectangle bounded by vertices $\boldsymbol{r}$ and $\boldsymbol{y}_i$.*

Definition 2. *Given a Pareto set $\mathcal{P}$ and reference point $\boldsymbol{r}$, the hypervolume improvement (HVI) of a set of points $\mathcal{Y}$ is: $\mathrm{HVI}(\mathcal{Y}, \mathcal{P}, \boldsymbol{r}) = \mathrm{HV}(\mathcal{P} \cup \mathcal{Y}, \boldsymbol{r}) - \mathrm{HV}(\mathcal{P}, \boldsymbol{r}).$*[1]

EHVI is the expectation of HVI over the posterior $\mathbb{P}(\boldsymbol{f}, \mathcal{D})$: $\alpha_{\mathrm{EHVI}}(\mathcal{X}_{\mathrm{cand}}) = \mathbb{E}\big[\mathrm{HVI}(\boldsymbol{f}(\mathcal{X}_{\mathrm{cand}})) \big]$. In the sequential setting, and assuming the objectives are independent and modeled with independent

---

[1]In this work, we omit the arguments $\mathcal{P}$ and $\boldsymbol{r}$ when referring to HVI for brevity.

GPs, EHVI can be expressed in closed form [69]. In other settings, EHVI can be approximated with MC integration. Following previous work, we assume that the reference point is known and specified by the decision maker [69] (see Appendix E.1.1 for additional discussion).

3.1 A review of hypervolume improvement computation using box decompositions

Definition 3. *For a set of objective vectors $\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q$, a reference point $\boldsymbol{r} \in \mathbb{R}^M$, and a non-dominated set $\mathcal{P}$, let $\Delta(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q, \mathcal{P}, \boldsymbol{r}) \subset \mathbb{R}^M$ denote the set of points (i) are dominated by $\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q$, dominate $\boldsymbol{r}$, and are not dominated by $\mathcal{P}$.*

Given $\mathcal{P}, \boldsymbol{r}$, the HVI of a new point $\boldsymbol{f}(\boldsymbol{x})$ is the HV of the intersection of space dominated by $\mathcal{P} \cup \{\boldsymbol{f}(\boldsymbol{x})\}$ and the non-dominated space. Figure 1b illustrates this for one new point $\boldsymbol{f}(\boldsymbol{x})$ for $M = 2$. The yellow region is $\Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})$ and the hypervolume improvement is the volume covered by $\Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})$. Since $\Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})$ is often a non-rectangular polytope, HVI is typically computed by partitioning the non-dominated space into disjoint axis-parallel rectangles [12, 68] (see Figure 1a) and using piece-wise integration [18].

Let $\{S_k\}_{k=1}^K$ be a partitioning the of non-dominated space into disjoint hyper-rectangles, where each S_k is defined by a pair of lower and upper vertices $\boldsymbol{l}_k \in \mathbb{R}^M$ and $\boldsymbol{u}_k \in \mathbb{R}^M \cup \{\boldsymbol{\infty}\}$. The high level idea is to sum the HV of $S_k \cap \Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})$ over all S_k. For each hyper-rectangle S_k, the intersection of S_k and $\Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})$ is a hyper-rectangle where the lower bound vertex is $\boldsymbol{l}_k$ and the upper bound vertex is the component-wise minimum of $\boldsymbol{u}_k$ and the new point $\boldsymbol{f}(\boldsymbol{x})$: $\boldsymbol{z}_k := \min\big[\boldsymbol{u}_k, \boldsymbol{f}(\boldsymbol{x})\big]$.
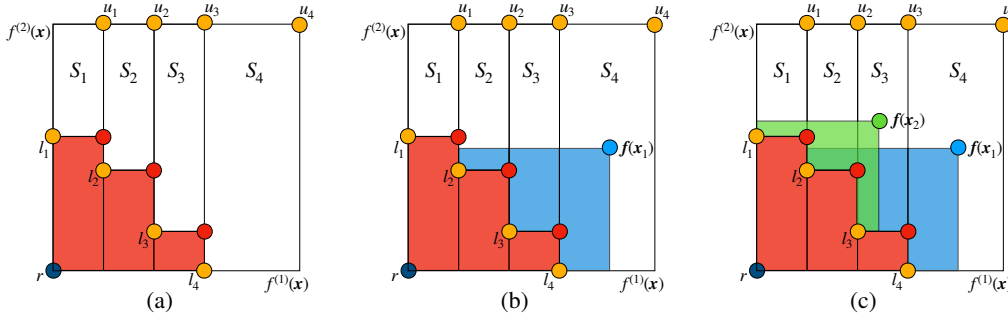

Figure 1: For M=2, (a) the dominated space (red) and the non-dominated space partitioned into disjoint boxes (white), (b) the HVI of one new point $\boldsymbol{f}(\boldsymbol{x})$, and (c) the HVI of two new points $\boldsymbol{f}(\boldsymbol{x}_1), \boldsymbol{f}(\boldsymbol{x}_2)$.

Hence, the HVI of a single outcome vector $\boldsymbol{f}(\boldsymbol{x})$ within S_k is given by $\text{HVI}_k\big(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{l}_k, \boldsymbol{u}_k\big) = \lambda_M\big(S_k \cap \Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})\big) = \prod_{m=1}^M \big[z_k^{(m)} - l_k^{(m)}\big]_+$, where $u_k^{(m)}, l_k^{(m)}, f^{(m)}(\boldsymbol{x})$, and $z_k^{(m)}$ denote the mth component of the corresponding vector and $[\cdot]_+$ denotes the $\min(\cdot, 0)$ operation. Summing over rectangles yields

$$\text{HVI}\big(\boldsymbol{f}(\boldsymbol{x})\big) = \sum_{k=1}^K \text{HVI}_k\big(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{l}_k, \boldsymbol{u}_k\big) = \sum_{k=1}^K \prod_{m=1}^M \big[z_k^{(m)} - l_k^{(m)}\big]_+ \quad (1)$$

3.2 Computing q-Hypervolume Improvement via the Inclusion-Exclusion Principle

Figure 1c illustrates the HVI in the $q = 2$ setting. Given q new points $\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q$, let $A_i := \Delta(\{\boldsymbol{f}(\boldsymbol{x}_i)\}, \mathcal{P}, \boldsymbol{r})$ for $i = 1, \ldots, q$ be the space dominated by $\boldsymbol{f}(\boldsymbol{x}_i)$ but not dominated by $\mathcal{P}$, independently of the other $q - 1$ points. Note that $\lambda_M(A_i) = \text{HVI}(\boldsymbol{f}(\boldsymbol{x}_i))$. The union of the subsets A_i is the space dominated jointly by the q new points: $\bigcup_{i=1}^q A_i = \bigcup_{i=1}^q \Delta(\{\boldsymbol{f}(\boldsymbol{x}_i)\}, \mathcal{P}, \boldsymbol{r})$, and the Lebesgue measure $\lambda_M\big(\bigcup_{i=1}^q A_i\big)$ is the joint HVI from the q new points. Since each subspace A_i is bounded, the restricted Lebesgue measure is finite and we may compute $\lambda_M\big(\bigcup_{i=1}^q A_i\big)$ using the inclusion-exclusion principle [13, 59]:

$$\text{HVI}(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q) = \lambda_M\bigg(\bigcup_{i=1}^q A_i\bigg) = \sum_{j=1}^q (-1)^{j+1} \sum_{1 \le i_1 \le \ldots \le i_j \le q} \lambda_M\big(A_{i_1} \cap \cdots \cap A_{i_j}\big) \quad (2)$$

4

Since $\{S_k\}_{k=1}^K$ is a disjoint partition, $\lambda_M(A_{i_1} \cap \cdots \cap A_{i_j}) = \sum_{k=1}^K \lambda_M(S_k \cap A_{i_1} \cap \cdots \cap A_{i_j})$, we can compute $\lambda_M(A_{i_1} \cap \cdots \cap A_{i_j})$ in a piece-wise fashion across the K hyper-rectangles $\{S_k\}_{k=1}^K$ as the HV of the intersection of $A_{i_1} \cap \cdots \cap A_{i_j}$ with each hyper-rectangle S_k. The inclusion-exclusion principle has been proposed for computing HV (not HVI) [45], but it is rarely used because complexity scales exponentially with the number of elements. However, the inclusion-exclusion principle is practical for computing the joint HVI of q points since typically $q << |\mathcal{P}|$.

This formulation has three advantages. First, while the new dominated space A_i can be a non-rectangular polytope, the intersection $A_i \cap S_k$ is a *rectangular* polytope, which simplifies computation of overlapping hypervolume. Second, the vertices defining the hyper-rectangle $S_k \cap A_{i_1} \cap \cdots \cap A_{i_j}$ are easily derived. The lower bound is simply the l_k lower bound of S_k, and the upper bound is the component-wise minimum $z_{k,i_1,...i_j} := \min\left[u_k, f(x_{i_1}), \ldots, f(x_{i_j})\right]$. Third, computation can be across all intersections of subsets $A_{i_1} \cap \cdots \cap A_{i_j}$ for $1 \le i_j \le \ldots \le i_j \le q$ and across all K hyper-rectangles can be performed in parallel. Explicitly, the HVI is computed as:

$$\text{HVI}(\{f(x_i)\}_{i=1}^q) = \sum_{k=1}^K \sum_{j=1}^q \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \prod_{m=1}^M \left[z_{k,X_j}^{(m)} - l_k^{(m)}\right]_+ \tag{3}$$

where $\mathcal{X}_j := \{X_j \subset \mathcal{X}_{\text{cand}} : |X_j| = j\}$ is the superset of all subsets of $\mathcal{X}_{\text{cand}}$ of size j, and $z_{k,X_j}^{(m)} := z_{k,i_1,...i_j}^{(m)}$ for $X_j = \{x_{i_1}, ..., x_{i_j}\}$. See Appendix A for further details of the derivation.

3.3 Computing *Expected* q-Hypervolume Improvement

The above approach for computing HVI assumes that we know the true objective values $f(\mathcal{X}_{\text{cand}}) = \{f(x_i)\}_{i=1}^q$. In BO, we instead compute qEHVI as the expectation over the posterior model posterior:

$$\alpha_{q\text{EHVI}}(\mathcal{X}_{\text{cand}}) = \mathbb{E}\left[\text{HVI}(f(\mathcal{X}_{\text{cand}}))\right] = \int_{-\infty}^{\infty} \text{HVI}(f(\mathcal{X}_{\text{cand}}))df. \tag{4}$$

Since no known analytical form is known [70] for $q > 1$ (or in the case of correlated outcomes), we estimate (4) using MC integration with samples from the joint posterior $\{f_t(x_i)\}_{i=1}^q \sim \mathbb{P}(f(x_1), ..., f(x_q)|\mathcal{D}), t = 1, \ldots N$. Let $z_{k,X_j,t}^{(m)} := \min\left[u_k, \min_{x' \in X_j} f_t(x')\right]$. Then,

$$\hat{\alpha}_{q\text{EHVI}}^N(\mathcal{X}_{\text{cand}}) = \frac{1}{N}\sum_{t=1}^N \text{HVI}(f_t(\mathcal{X}_{\text{cand}})) = \frac{1}{N}\sum_{t=1}^N \sum_{k=1}^K \sum_{j=1}^q \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \prod_{m=1}^M [z_{k,X_j,t}^{(m)} - l_k^{(m)}]_+ \tag{5}$$

Provided that $\{S_k\}_{k=1}^K$ is an exact partitioning, (5) is an *exact* computation of qEHVI up to the MC estimation error, which scales as $1/\sqrt{N}$ when using iid MC samples regardless of the dimension of the search space [18]. In practice, we use randomized quasi MC methods [8] to reduce the variance and empirically observe low estimation error (see Figure 5a in the Appendix for a comparison of analytic EHVI and (quasi-)MC-based qEHVI).

qEHVI requires computing the volume of $2^q - 1$ hyper-rectangles (the number of subsets of q) for each of K hyper-rectangles and N MC samples. Given posterior samples, the time complexity on a single-threaded machine is: $T_1 = O(MNK(2^q - 1))$. In the two-objective case, $K = |\mathcal{P}| + 1$, but K is super-polynomial in M [68]. The number of boxes required for a decomposition of the non-dominated space is unknown for $M \ge 4$ [68]. qEHVI is agnostic to the partitioning algorithm used, and in F.4, we demonstrate using qEHVI in higher-dimensional objective spaces using an approximate box decomposition algorithm [11]. Despite the daunting workload, the critical work path—the time complexity of the smallest non-parallelizable unit—is constant: $T_\infty = O(1)$.[2] On highly-threaded many-core hardware (e.g. GPUs), our formulation achieves tractable wall times in many practical scenarios: as is shown in Figure 11 in the Appendix, the computation time is nearly constant with increasing q until an inflection point at which the workload saturates the available cores. For additional discussion of both time and memory complexity of qEHVI see Appendix A.4.

3.4 Outcome Constraints

Our proposed qEHVI acquisition function is easily extended to constraints on auxiliary outcomes. We consider the scenario where we receive observations of M objectives $f(x) \in \mathbb{R}^M$ and V constraints

---

[2]As evident from (5), the critical path consists of 3 multiplications and 5 summations.

$c^{(v)} \in \mathbb{R}^V$, all of which are assumed to be "black-box". We assume w.l.o.g. that $c^{(v)}$ is feasible iff $c^{(v)} \geq 0$. In the constrained optimization setting, we aim to identify the feasible Pareto set: $\mathcal{P}_{\text{feas}} = \{f(x) \ s.t. \ c(x) \geq 0, \ \nexists \ x' : c(x') \geq 0, \ f(x') \succ f(x)\}$. The natural improvement measure in the constrained setting is *feasible* HVI, which we define for a single candidate point x as $\text{HVI}_c(f(x), c(x)) := \text{HVI}[f(x)] \cdot \mathbb{1}[c(x) \geq 0]$. Taking expectations, the constrained expected HV can be seen to be the HV weighted by the probability of feasibility. In Appendix A.3, we detail how performing feasibility-weighting on the sample-level allows us to include such auxiliary outcome constraints into our MC formulation in a straightforward way.

4 Optimizing q-Expected Hypervolume Improvement

4.1 Differentiability

While an analytic formula for the gradient of EHVI exists for the $M = 2$ objective case in the unconstrained, sequential ($q = 1$) setting, no such formula is known in 1) the case of $M > 2$ objectives, 2) the constrained setting, and 3) for $q > 1$. Leveraging the re-parameterization trick [39, 64] and auto-differentiation, we are able to automatically compute exact gradients of the MC-estimator qEHVI in *all* of the above settings, as well as the gradient of analytic EHVI for $M \geq 2$ (see Figure 5b in the Appendix for a comparison of the exact gradients of EHVI and the sample average gradients of qEHVI for $M = 3$).[3,4]

4.2 Optimization via Sample Average Approximation

We show in Appendix C that if mean and covariance function of the GP are sufficiently regular, the gradient of the MC estimator (5) is an unbiased estimate of the gradient of the exact acquisition function (4). To maximize qEHVI, we could therefore directly apply stochastic optimization methods, as has previously been done for single-outcome acquisition functions [64, 66]. Instead, we opt to use the sample average approximation (SAA) approach from Balandat et al. [5], which allows us to employ deterministic, higher-order optimizers to achieve faster convergence rates. Informally (see Appendix C for the formal statement), if $\hat{x}_N^* \in \arg\max_{x \in \mathcal{X}} \hat{\alpha}_{q\text{EHVI}}^N(x)$, we can show under some regularity conditions that, as $N \to \infty$, (i) $\hat{\alpha}_{q\text{EHVI}}^N(\hat{x}_N^*) \to \max_{x \in \mathcal{X}} \alpha_{q\text{EHVI}}(x) \ a.s.$, and (ii) $\text{dist}(\hat{x}_N^*, \arg\max_{x \in \mathcal{X}} \alpha_{q\text{EHVI}}(x)) \to 0 \ a.s.$. These results hold for any covariance function satisfying the regularity conditions, including such ones that model correlation between outcomes. In particular, our results do not require the outputs to be modeled by independent GPs.

Figure 2a demonstrates the importance of using exact gradients for efficiently and effectively optimizing EHVI and qEHVI by comparing the following optimization methods: L-BFGS-B with exact gradients, L-BFGS-B with gradients approximated via finite differences, and CMA-ES (without gradients). The cumulative time spent optimizing the acquisition function is an order of magnitude less when using exact gradients rather than approximate gradients or zeroth order methods.

4.3 Sequential Greedy and Joint Batch Optimization

Jointly optimizing q candidates increases in difficulty with q because the problem dimension is dq. An alternative is to sequentially and greedily select candidates and condition the acquisition function on the previously selected pending points when selecting the next point [65]. Using a submodularity argument similar to that in Wilson et al. [64], the sequential greedy approximation of qEHVI enjoys regret of no more than $\frac{1}{e} \alpha_{q\text{EHVI}}^*$, where $\alpha_{q\text{EHVI}}^*$ is the optima of $\alpha_{q\text{EHVI}}$ [23] (see Appendix B).

Although sequential greedy approaches have been considered for many acquisition functions [65], no previous work has proposed a proper sequential greedy approach (with integration over the posterior) for parallel EHVI, as this would require computing the Pareto front under each sample f_t from the joint posterior before computing the hypervolume improvement. These operations would be computationally expensive for even modest N and non-differentiable. qEHVI avoids determining the Pareto set for each sample by using inclusion-exclusion principle to compute the joint HVI over the pending points $x_1, ..., x_{i-1}$ and the new candidate x_i for each MC sample. Figure 2b empirically

---

[3]Technically, min and max are only sub-differentiable, but are known to be well-behaved [64]. In our MC setting with GP posteriors, qEHVI is differentiable w.p. 1 if x contains no repeated points.

[4]For the constrained case, we replace the indicator with a differentiable sigmoid approximation.
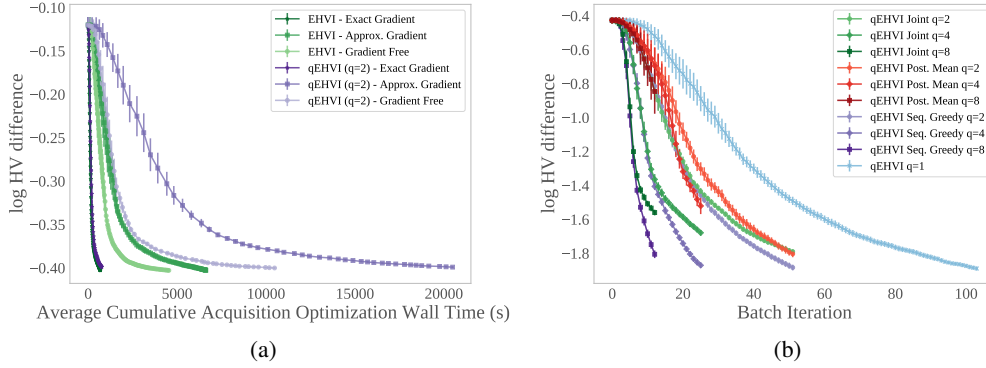
Figure 2: (a) A comparison of EHVI and qEHVI ($q = 2$) optimized with L-BFGS-B using exact gradients, L-BFGS-B using gradients approximated using finite differences, and CMA-ES, a gradient-free method. (b) A comparison of joint optimization, sequential greedy optimization with proper integration at the pending points, and sequential greedy using the posterior mean. Both plots show optimization performance on a DTLZ2 problem ($d = 6, M = 2$) with a budget of 100 evaluations (plus the initial quasi-random design). We report means and 2 standard errors across 20 trials.

demonstrates the improved optimization performance from properly integrating over the unobserved outcomes rather than using the posterior mean or jointly optimizing the q candidates.

5   Benchmarks

We empirically evaluate qEHVI on synthetic and real world optimization problems. We compare qEHVI[5] against existing state-of-the-art methods including SMS-EGO[6], PESMO[6], TS-TCH[5], and analytic EHVI [68] with gradients[5]. Additionally, we compare against a novel extension of ParEGO [40] that supports parallel evaluation and constraints (neither of which have been done before to our knowledge); we call this method qPAREGO[5]. Additionally, we include a quasi-random baseline that selects candidates from a scrambled Sobol sequence. See Appendix E.1 for details on all baseline algorithms.

Synthetic Benchmarks   We evaluate optimization performance on four benchmark problems in terms of log hypervolume difference, which is defined as the difference between the hypervolume of the true (feasible) Pareto front and the hypervolume of the approximate (feasible) Pareto front based on the observed data; in the case that the true Pareto front is unknown (or not easily approximated), we evaluate the hypervolume indicator. All references points and search spaces are provided in Appendix E.2. For synthetic problems, we consider the Branin-Currin problem ($d = 2, M = 2$, convex Pareto front) [6] and the C2-DTLZ2 ($d = 12, M = 2, V = 1$, concave Pareto front), which is a standard constrained benchmark from the MO literature [16] (see Appendix F.1 for additional synthetic benchmarks).

Real-World Benchmarks
Structural Optimization in Automobile Safety Design (VEHICLESAFETY): Vehicle crash safety is an important consideration in the structural design of automobiles. A lightweight car is preferable because of its potentially lower manufacturing cost and better fuel economy, but lighter material can fare worse than sturdier alternatives in a collision, potentially leading to increased vehicle damage and more severe injury to the vehicle occupants [72]. We consider the problem designing the thickness of 5 reinforced parts of the frontal frame of a vehicle that considerably affect crash safety. The goal is to minimize: 1) the *mass* of the vehicle; 2) the *collision acceleration* in a full frontal crash—a proxy for bio-mechanical trauma to the vehicle occupants from the acceleration; and 3) the *toe-board*

---

[5]Acquisition functions are available as part of the open-source library BoTorch [5]. Code is available at `https://github.com/pytorch/botorch`.

[6]We leverage existing implementations from the Spearmint library. The code is available at `https://github.com/HIPS/Spearmint/tree/PESM`.
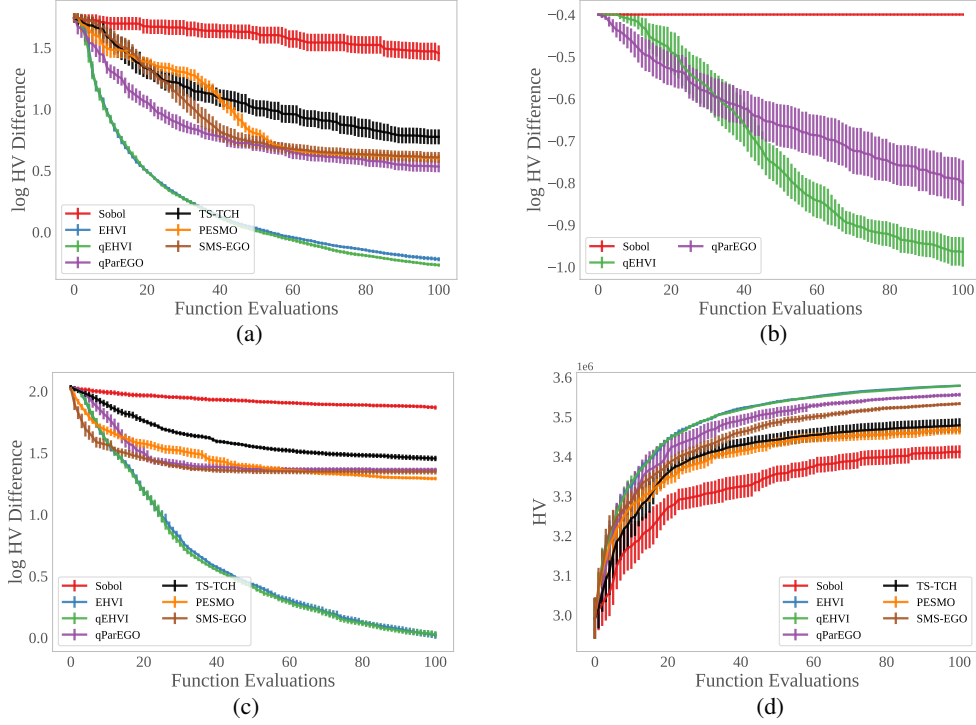
Figure 3: Sequential optimization performance on (a) on the Branin-Currin problem ($q = 1$), (b) the C2-DTLZ2 problem, (c) the vehicle crash safety problem ($q = 1$), and (d) the ABR control problem ($q = 1$). We report the means and 2 standard errors across 20 trials.

intrusion—a measure of the most extreme mechanical damage to the vehicle in an off-frontal collision [44]. For this problem, we optimize the surrogate from Tanabe and Ishibuchi [60].

Policy Optimization for Adaptive Bitrate Control (ABR): Many web services adapt video playback quality adaptively based on the receiver's network bandwith to maintain steady, high quality stream with minimal stalls and buffer periods [47]. Previous works have proposed controllers with different scalarized objective functions [46], but in many cases, engineers may prefer to learn the set of optimal trade-offs between their metrics of interest, rather than specifying a scalarized objective in advance. In this problem, we decompose the objective function proposed in Mao et al. [46] into its constituent metrics and optimize 4 parameters of an ABR control policy on the Park simulator [48] to maximize video quality (bitrate) and minimize stall time. See Appendix E.2 for details.

5.1 Results

Figure 3 shows that qEHVI outperforms all baselines in terms of sequential optimization performance on all evaluated problems. Table 1 shows that qEHVI achieves wall times that are an order of magnitude smaller than those of PESMO on a CPU in sequential optimization, and maintains competitive wall times even relative to qPAREGO (which has a significantly smaller workload) for large q on a GPU. TS-TCH has by far the fastest wall time, but this comes at the cost of inferior optimization performance.

Figure 4 illustrates optimization performance of *parallel* acquisition functions for varying batch sizes. Increasing the level of parallelism leads to faster convergence for all algorithms (Figure 4a). In contrast with other algorithms, qEHVI's sample complexity does not deteriorate substantially when high levels of parallelism are used (Figure 4b).

8

Table 1: Acquisition Optimization wall time in seconds on a CPU (2x Intel Xeon E5-2680 v4 @ 2.40GHz) and a GPU (Tesla V100-SXM2-16GB). We report the mean and 2 standard errors across 20 trials. NA indicates that the algorithm does not support constraints.

CPU	BRANINCURRIN	C2DTLZ2	ABR	VEHICLESAFETY
PESMO (q=1)	249.16 ($\pm$19.35)	NA	214.16 ($\pm$18.38)	492.64 ($\pm$58.98)
SMS-EGO (q=1)	146.1 ($\pm$8.57)	NA	89.54 ($\pm$5.79)	115.11 ($\pm$8.21)
TS-TCH (q=1)	2.82 ($\pm$0.03)	NA	17.22 ($\pm$0.04)	47.46 ($\pm$0.05)
qPAREGO (q=1)	1.56 ($\pm$0.16)	4.01 ($\pm$0.77)	7.47 ($\pm$0.67)	1.74 ($\pm$0.27)
EHVI (q=1)	3.04 ($\pm$0.16)	NA	2.48 ($\pm$0.19)	15.18 ($\pm$2.24)
qEHVI (q=1)	3.63 ($\pm$0.23)	5.4 ($\pm$1.18)	6.15 ($\pm$0.71)	67.54 ($\pm$10.45)

| GPU	BRANINCURRIN	C2DTLZ2	ABR	VEHICLESAFETY
TS-TCH (q=1)	0.07 ($\pm$0.00)	NA	0.16 ($\pm$0.00)	0.32 ($\pm$0.0)
TS-TCH (q=2)	0.07 ($\pm$0.00)	NA	0.15 ($\pm$0.00)	0.34 ($\pm$0.01)
TS-TCH (q=4)	0.09 ($\pm$0.01)	NA	0.15 ($\pm$0.00)	0.31 ($\pm$0.01)
TS-TCH (q=8)	0.08 ($\pm$0.00)	NA	0.16 ($\pm$0.00)	0.34 ($\pm$0.01)
qPAREGO (q=1)	3.2 ($\pm$0.37)	3.85 ($\pm$0.91)	9.64 ($\pm$0.96)	3.44 ($\pm$0.51)
qPAREGO (q=2)	7.12 ($\pm$0.81)	12.1 ($\pm$2.77)	21.19 ($\pm$1.53)	7.32 ($\pm$0.97)
qPAREGO (q=4)	15.34 ($\pm$1.69)	39.71 ($\pm$7.40)	35.46 ($\pm$2.32)	17.2 ($\pm$2.29)
qPAREGO (q=8)	32.11 ($\pm$4.14)	99.58 ($\pm$15.20)	72.52 ($\pm$5.04)	39.72 ($\pm$7.13)
EHVI (q=1)	4.53 ($\pm$0.23)	NA	6.82 ($\pm$0.55)	8.95 ($\pm$0.64)
qEHVI (q=1)	5.98 ($\pm$0.28)	3.36 ($\pm$0.94)	7.71 ($\pm$0.67)	10.43 ($\pm$0.64)
qEHVI (q=2)	11.37 ($\pm$0.56)	21.56 ($\pm$3.45)	18.32 ($\pm$1.48)	17.67 ($\pm$1.54)
qEHVI (q=4)	25.29 ($\pm$1.51)	89.18 ($\pm$10.86)	44.44 ($\pm$3.53)	54.25 ($\pm$4.17)
qEHVI (q=8)	102.46 ($\pm$9.22)	215.74 ($\pm$15.85)	100.64 ($\pm$7.22)	255.72 ($\pm$23.73)

(a)   (b)
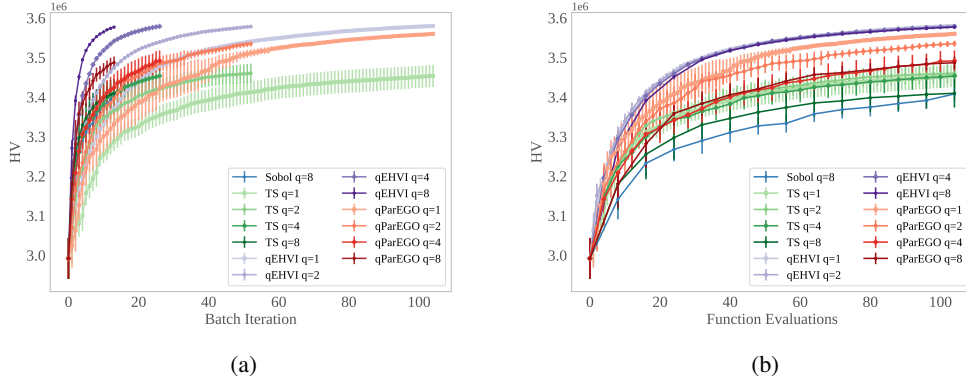
Figure 4: Parallel optimization performance on the ABR problem with varying batch sizes (q) by (a) *batch BO iterations* and (b) *function evaluations*.

6 Discussion

We present a practical and efficient acquisition function, qEHVI, for parallel, constrained multi-objective Bayesian optimization. Leveraging differentiable programming, modern parallel hardware, and the Sample Average Approximation, we efficiently optimize qEHVI via quasi second-order methods and provide theoretical convergence guarantees for our approach. Empirically, we demonstrate that our method out-performs state-of-the-art multi-objective Bayesian optimization methods.

One limitation of our approach is that it currently assumes noiseless observations, which, to our knowledge, is the case with all formulations of EHVI. Integrating over the uncertainty around the previous observations [43] by using MC samples over the new candidates and the training points, one may be able to account for the noise.Another limitation of qEHVI is that its scalability is limited the partitioning algorithm, precluding its use in high-dimensional objective spaces. More scalable partitioning algorithms, either approximate algorithms (e.g. the algorithm proposed by Couckuyt et al. [11], which we examine briefly in Appendix F.4) or more efficient exact algorithms that result in fewer disjoint hyper-rectangles (e.g. [41, 17, 69]), will improve the scalability and computation time of of qEHVI. We hope this work encourages researchers to consider more improvements from applying modern computational paradigms and tooling to Bayesian optimization.

7 Statement of Broader Impact

Optimizing a single outcome commonly comes at the expense of other secondary outcomes. In some cases, decision makers may be able to form a scalarization of their objectives in advance, but in the researcher's experience, formulating such trade-offs in advance is difficult for most. Improvements to the optimization performance and practicality of multi-objective Bayesian optimization have the potential to allow decision makers to better understand and make more informed decisions across multiple trade-offs. We expect these directions to be particularly important as Bayesian optimization is increasingly used for applications such as recommender systems [42], where auxiliary goals such as fairness must be accounted for. Of course, at the end of the day, exactly what objectives decision makers choose to optimize, and how they balance those trade-offs (and whether that is done in equitable fashion) is up to the individuals themselves.

Acknowledgments

References

[1] M. Abdolshah, A. Shilton, S. Rana, S. Gupta, and S. Venkatesh. Expected hypervolume improvement with constraints. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3238–3243, 2018.

[2] Arash Asadpour, Hamid Nazerzadeh, and Amin Saberi. Stochastic submodular maximization. In Christos Papadimitriou and Shuzhong Zhang, editors, *Internet and Network Economics*. Springer Berlin Heidelberg, 2008.

[3] R. Astudillo and P. Frazier. Bayesian optimization of composite functions. *Forthcoming, in Proceedings of the 35th International Conference on Machine Learning*, 2019.

[4] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Theory of the hypervolume indicator: Optimal mu-distributions and the choice of the reference point. In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, FOGA '09, page 87–102, New York, NY, USA, 2009. Association for Computing Machinery.

[5] 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.

[6] 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.

[7] Eric Bradford, Artur Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm. *Journal of Global Optimization*, 71, 02 2018. doi: 10.1007/s10898-018-0609-2.

[8] Russel E Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49, 1998.

[9] Mauro Cerasoli and Aniello Fedullo. The inclusion-exclusion principle. *Journal of Interdisciplinary Mathematics*, 5(2):127–141, 2002.

[10] Anirban Chaudhuri, Raphael Haftka, Peter Ifju, Kelvin Chang, Christopher Tyler, and Tony Schmitz. Experimental flapping wing optimization and uncertainty quantification using limited samples. *Structural and Multidisciplinary Optimization*, 51, 11 2014. doi: 10.1007/s00158-014-1184-x.

[11] I. Couckuyt, D. Deschrijver, and T. Dhaene. Towards efficient multiobjective optimization: Multiobjective statistical criterions. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, 2012.

[12] 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.

[13] Daniel A. da Silva. Proprietades geraes. *J. de l'Ecole Polytechnique*, cah. 30. I, 1854.

[14] 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.

[15] Kalyan Deb, L. Thiele, Marco Laumanns, and Eckart Zitzler. Scalable multi-objective optimization test problems. volume 1, pages 825–830, 06 2002. ISBN 0-7803-7282-4. doi: 10.1109/CEC.2002.1007032.

[16] Kalyanmoy Deb. *Constrained Multi-objective Evolutionary Algorithm*, pages 85–118. Springer International Publishing, Cham, 2019.

[17] Kerstin Dächert, Kathrin Klamroth, Renaud Lacour, and Daniel Vanderpooten. Efficient computation of the search region in multi-objective optimization. *European Journal of Operational Research*, 260(3):841 – 855, 2017.

[18] 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.

[19] 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)*, pages 2147–2154, 2011.

[20] Michael Emmerich, Kaifeng Yang, André Deutz, Hao Wang, and Carlos M. Fonseca. *A Multicriteria Generalization of Bayesian Global Optimization*, pages 229–242. Springer International Publishing, 2016.

[21] Michael T. M. Emmerich and Carlos M. Fonseca. Computing hypervolume contributions in low dimensions: Asymptotically optimal algorithm and complexity results. In Ricardo H. C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Greco, editors, *Evolutionary Multi-Criterion Optimization*, pages 121–135, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[22] Paul Feliot, Julien Bect, and Emmanuel Vazquez. A bayesian approach to constrained single- and multi-objective optimization. *Journal of Global Optimization*, 67(1-2):97–133, Apr 2016. ISSN 1573-2916. doi: 10.1007/s10898-016-0427-3. URL http://dx.doi.org/10.1007/s10898-016-0427-3.

[23] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. *An analysis of approximations for maximizing submodular set functions—II*, pages 73–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

[24] Tobias Friedrich and Frank Neumann. Maximizing submodular functions under matroid constraints by multi-objective evolutionary algorithms. In Thomas Bartz-Beielstein, Jürgen Branke, Bogdan Filipič, and Jim Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, pages 922–931, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10762-2.

[25] 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.

[26] Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. Predictive entropy search for multi-objective bayesian optimization with constraints. *Neurocomputing*, 361:50–68, 2019.

[27] Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. Parallel predictive entropy search for multi-objective bayesian optimization with constraints, 2020.

[28] David Gaudrie, Rodolphe Le Riche, Victor Picheny, Benoît Enaux, and Vincent Herbert. Targeting solutions in bayesian multi-objective optimization: sequential and batch versions. *Annals of Mathematics and Artificial Intelligence*, 88(1-3):187–212, Aug 2019. ISSN 1573-7470. doi: 10.1007/s10472-019-09644-8. URL http://dx.doi.org/10.1007/s10472-019-09644-8.

[29] Michael A. Gelbart, Jasper Snoek, and Ryan P. Adams. Bayesian optimization with unknown constraints. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, UAI, 2014.

[30] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. *Kriging Is Well-Suited to Parallelize Optimization*, pages 131–162. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[31] P. Glasserman. Performance continuity and differentiability in monte carlo optimization. In *1988 Winter Simulation Conference Proceedings*, pages 518–524, 1988.

[32] Nikolaus Hansen. *The CMA Evolution Strategy: A Comparing Review*, volume 192, pages 75–102. 06 2007. doi: 10.1007/3-540-32494-1_4.

[33] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, Amar Shah, and Ryan P. Adams. Predictive entropy search for multi-objective bayesian optimization, 2015.

[34] Iris Hupkens, Andre Deutz, Kaifeng Yang, and Michael Emmerich. Faster exact algorithms for computing expected hypervolume improvement. In Antonio Gaspar-Cunha, Carlos Henggeler Antunes, and Carlos Coello Coello, editors, *Evolutionary Multi-Criterion Optimization*, pages 65–79. Springer International Publishing, 2015.

[35] 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. URL https://doi.org/10.1145/2001576.2001666.

[36] 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):411–440, September 2018.

[37] Donald Jones, C. Perttunen, and B. Stuckman. Lipschitzian optimisation without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79:157–181, 01 1993. doi: 10.1007/BF00941892.

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

[39] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114, Dec 2013.

[40] 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.

[41] Renaud Lacour, Kathrin Klamroth, and Carlos M. Fonseca. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, 79:347 – 360, 2017.

[42] Benjamin Letham and Eytan Bakshy. Bayesian optimization for policy search via online-offline experimentation. *Journal of Machine Learning Research*, 20(145):1–30, 2019. URL http://jmlr.org/papers/v20/18-225.html.

[43] Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 06 2019. doi: 10.1214/18-BA1110.

[44] 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*, 35:561–569, 06 2008. doi: 10.1007/s00158-007-0163-x.

[45] 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.

[46] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, page 197–210, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346535. doi: 10.1145/3098822.3098843. URL https://doi.org/10.1145/3098822.3098843.

[47] Hongzi Mao, Shannon Chen, Drew Dimmery, Shaun Singh, Drew Blaisdell, Yuandong Tian, Mohammad Alizadeh, and Eytan Bakshy. Real-world video adaptation with reinforcement learning. 2019.

[48] Hongzi Mao, Parimarjan Negi, Akshay Narayan, Hanrui Wang, Jiacheng Yang, Haonan Wang, Ryan Marcus, Ravichandra Addanki, Mehrdad Khani Shirkoohi, Songtao He, Vikram Nathan, Frank Cangialosi, Shaileshh Bojja Venkatakrishnan, Wei-Hung Weng, Shu-Wen Han, Tim Kraska, and Mohammad Alizadeh. Park: An open platform for learning-augmented computer systems. In *NeurIPS*, 2019.

[49] Sébastien Marmin, Clément Chevalier, and David Ginsbourger. Differentiating the multipoint expected improvement for optimal batch design. In Panos Pardalos, Mario Pavone, Giovanni Maria Farinella, and Vincenzo Cutello, editors, *Machine Learning, Optimization, and Big Data*, pages 37–48, Cham, 2015. Springer International Publishing.

[50] B. Paria, K. Kandasamy, and B. Póczos. A Flexible Multi-Objective Bayesian Optimization Approach using Random Scalarizations. *ArXiv e-prints*, May 2018.

[51] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 2017.

[52] Victor Picheny. Multiobjective optimization using gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 25, 10 2013. doi: 10.1007/s11222-014-9477-x.

[53] 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.

[54] Alma A. M. Rahat, Richard M. Everson, and Jonathan E. Fieldsend. Alternative infill strategies for expensive multi-objective optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, page 873–880, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349208.

[55] Carl Edward Rasmussen. *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[56] Jerry Segercrantz. Inclusion-exclusion and characteristic functions. *Mathematics Magazine*, 71(3):216–218, 1998. ISSN 0025570X, 19300980. URL http://www.jstor.org/stable/2691209.

[57] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[58] 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*, ICML'10, page 1015–1022, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

[59] J. Sylvester. Note sur la théorème de legendre. *Comptes Rendus Acad. Sci.*, 96:463–465, 1883.

[60] Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2020.106078.

[61] 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.

[62] Takashi Wada and Hideitsu Hino. Bayesian optimization for multi-objective optimization and multi-point search, 2019.

[63] Jialei Wang, Scott C. Clark, Eric Liu, and Peter I. Frazier. Parallel bayesian global optimization of expensive functions, 2016.

[64] J. T. Wilson, R. Moriconi, F. Hutter, and M. P. Deisenroth. The reparameterization trick for acquisition functions. *ArXiv e-prints*, December 2017.

[65] 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.

[66] Jian Wu and Peter I. Frazier. The parallel knowledge gradient method for batch bayesian optimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3134–3142, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[67] Kaifeng Yang, Michael Emmerich, André Deutz, and Carlos M. Fonseca. Computing 3-d expected hypervolume improvement and related integrals in asymptotically optimal time. In *9th International Conference on Evolutionary Multi-Criterion Optimization - Volume 10173*, EMO 2017, page 685–700, Berlin, Heidelberg, 2017. Springer-Verlag.

[68] Kaifeng Yang, Michael Emmerich, André H. Deutz, and Thomas Bäck. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *CoRR*, abs/1904.12672, 2019.

[69] 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. doi: https://doi.org/10.1016/j.swevo.2018.10.007. URL http://www.sciencedirect.com/science/article/pii/S2210650217307861.

[70] Kaifeng Yang, Pramudita Palar, Michael Emmerich, Koji Shimoyama, and Thomas Bäck. A multi-point mechanism of expected hypervolume improvement for parallel multi-objective bayesian global optimization. pages 656–663, 07 2019. doi: 10.1145/3321707.3321784.

[71] Kaifeng Yang, Pramudita Satria Palar, Michael Emmerich, Koji Shimoyama, and Thomas Bäck. A multi-point mechanism of expected hypervolume improvement for parallel multi-objective bayesian global optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, page 656–663, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361118. doi: 10.1145/3321707.3321784. URL `https://doi.org/10.1145/3321707.3321784`.

[72] R. J. Yang, N. Wang, C. H. Tho, J. P. Bobineau, and B. P. Wang. Metamodeling Development for Vehicle Frontal Impact Simulation. *Journal of Mechanical Design*, 127(5):1014–1020, 01 2005.

[73] 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.

Appendix to:

Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization

A Derivation of q-Expected Hypervolume Improvement

A.1 Hypervolume Improvement via the Inclusion-Exclusion Principle

The hypervolume improvement of $\boldsymbol{f}(\boldsymbol{x})$ within the hyper-rectangle S_k is the volume of $S_k \cap \Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})$ and is given by:

$$\text{HVI}_k\big(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{l}_k, \boldsymbol{u}_k\big) = \lambda_M\big(S_k \cap \Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})\big) = \prod_{m=1}^{M} \big[z_k^{(m)} - l_k^{(m)}\big]_+,$$

where $u_k^{(m)}, l_k^{(m)}, f^{(m)}(\boldsymbol{x})$, and $z_k^{(m)}$ denote the mth component of the corresponding vector and $[\cdot]_+$ denotes the $\min(\cdot, 0)$ operation. Summing over all S_k gives the total hypervolume improvement:

$$\begin{aligned}
\text{HVI}\big(\boldsymbol{f}(\boldsymbol{x})\big) &= \sum_{k=1}^{K} \text{HVI}_k\big(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{l}_k, \boldsymbol{u}_k\big) \\
&= \sum_{k=1}^{K} \lambda_M\big(S_k \cap \Delta(\{\boldsymbol{f}(\boldsymbol{x})\}, \mathcal{P}, \boldsymbol{r})\big) \\
&= \sum_{k=1}^{K} \prod_{m=1}^{M} \big[z_k^{(m)} - l_k^{(m)}\big]_+.
\end{aligned}$$

We can extend the HVI computation to the $q > 1$ case using the inclusion-exclusion principle.

Principle 1. *The inclusion-exclusion principle* [13, 59, 9] *Given a finite measure space* $(B, \mathcal{A}, \mu)$ *and a finite sequence of potentially empty or overlapping sets* $\{A_i\}_i = 1^n$ *where* $A_i \in \mathcal{A}$ *and* $\mu(B) < \infty$, *then,*

$$\lambda_M\bigg(\bigcup_{i=1}^{p} A_i\bigg) = \sum_{j=1}^{p} (-1)^{j+1} \sum_{1 \leq i_1 \leq \ldots \leq i_j \leq p} \lambda_M\big(A_{i_1} \cap \ldots \cap A_{i_j}\big)$$

In the context of computing the joint HVI of q new points $\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^{q}$, each subset A_i for $i = 1, \ldots, q$ is the set of points contained in $\Delta(\{\boldsymbol{f}(\boldsymbol{x}_i)\}, \mathcal{P}, \boldsymbol{r})$ — independently of the other $q - 1$ points. $\lambda_M(A_i)$ is the hypervolume improvement from the new point $\boldsymbol{f}(\boldsymbol{x}_i)$: $\lambda_M(A_i) = \text{HVI}(\boldsymbol{f}(\boldsymbol{x}_i))$. The union of these subsets is the set of points in the new space dominated by the q new points: $\bigcup_{i=1}^{q} A_i = \bigcup_{i=1}^{q} \Delta(\{\boldsymbol{f}(\boldsymbol{x}_i)\}, \mathcal{P}, \boldsymbol{r})$. The hypervolume of $\bigcup_{i=1}^{q} \Delta(\{\boldsymbol{f}(\boldsymbol{x}_i)\}, \mathcal{P}, \boldsymbol{r})$ is the hypervolume improvement from the q new points:

$$\begin{aligned}
\text{HVI}(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^{q}) &= \lambda_M\bigg(\bigcup_{i=1}^{q} A_i\bigg) \\
&= \sum_{j=1}^{q} (-1)^{j+1} \sum_{1 \leq i_1 \leq \ldots \leq i_j \leq q} \lambda_M\big(A_{i_1} \cap \cdots \cap A_{i_j}\big)
\end{aligned}$$

To compute $\lambda_M\big(A_{i_1} \cap \cdots \cap A_{i_j}\big)$, we partition the space covered by $A_{i_1} \cap \cdots \cap A_{i_j}$ across the K hyper-rectangles $\{S_k\}_{k=1}^{K}$ and compute the hypervolume of the overlapping space of $A_{i_1} \cap \cdots \cap A_{i_j}$ with each S_k independently. Since $\{S_k\}_{k=1}^{K}$ is a disjoint partition, summing over K gives the hypervolume of $A_{i_1} \cap \cdots \cap A_{i_j}$:

$$\lambda_M\big(A_{i_1} \cap \cdots \cap A_{i_j}\big) = \sum_{k=1}^{K} \lambda_M\big(S_k \cap A_{i_1} \cap \cdots \cap A_{i_j}\big)$$

This has two advantages. First, the new dominated space A_i can be a non-rectangular polytope, but the intersection $A_i \cap S_k$ is a *rectangular* polytope, which simplifies computation of overlapping hypervolume.

Second, the vertices defining the hyper-rectangle encapsulated by $S_k \cap A_{i_1} \cap \cdots \cap A_{i_j}$ are easily derived. The lower bound is simply the $\boldsymbol{l}_k$ lower bound of S_k and the upper bound is the component-wise minimum $\boldsymbol{z}_{k,i_1,\ldots i_j} = \min\left[\boldsymbol{u}_k, \boldsymbol{f}(\boldsymbol{x}_{i_1}), \ldots, \boldsymbol{f}(\boldsymbol{x}_{i_j})\right]$.

Importantly, this is computationally tractable because this specific approach enables parallelizing computation across all intersections of subsets $A_{i_1} \cap \cdots \cap A_{i_j}$ for $1 \leq i_j \leq \ldots \leq i_j \leq q$ and across all K hyper-rectangles. Explicitly, the HVI is computed as:

$$
\begin{aligned}
\mathrm{HVI}(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q) &= \lambda_M\left(\bigcup_{i=1}^p A_i\right) \\
&= \sum_{j=1}^q \sum_{1 \leq i_1 \leq \ldots \leq i_j \leq q} (-1)^{j+1} \lambda_M\left(A_{i_1} \cap \cdots \cap A_{i_j}\right) \\
&= \sum_{k=1}^K \sum_{j=1}^q \sum_{1 \leq i_1 \leq \ldots \leq i_j \leq q} (-1)^{j+1} \lambda_M\left(S_k \cap A_{i_1} \cap \cdots \cap A_{i_j}\right) \\
&= \sum_{k=1}^K \sum_{j=1}^q \sum_{1 \leq i_1 \leq \ldots \leq i_j \leq q} (-1)^{j+1} \lambda_M\left(S_k \cap \Delta(\{\boldsymbol{f}(\boldsymbol{x}_{i_1})\}, \mathcal{P}, \boldsymbol{r}) \cap \ldots \cap \Delta(\{\boldsymbol{f}(\boldsymbol{x}_{i_j})\}, \mathcal{P}, \boldsymbol{r})\right) \\
&= \sum_{k=1}^K \sum_{j=1}^q \sum_{1 \leq i_1 \leq \ldots \leq i_j \leq q} (-1)^{j+1} \prod_{m=1}^M \left[z_{k,i_1,\ldots i_j}^{(m)} - l_k^{(m)}\right]_+ \\
&= \sum_{k=1}^K \sum_{j=1}^q \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \prod_{m=1}^M \left[z_{k,X_j}^{(m)} - l_k^{(m)}\right]_+
\end{aligned}
$$

where X_j is the superset all subsets of $\mathcal{X}_{\mathrm{cand}}$ of size j: $\mathcal{X}_j = \{X_j \subset \mathcal{X}_{\mathrm{cand}} : |X_j| = j\}$ and $z_{k,X_j}^{(m)} = z_{k,i_1,\ldots i_j}^{(m)}$ for $X_j = \{\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_j}\}$.

A.2 Computing *Expected* Hypervolume Improvement

The above approach for computing HVI assumes we know the true objective values $\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q$. Since we do not know the true function values $\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q$, we compute qEHVI as the expectation over the GP posterior.

$$
\alpha_{q\mathrm{EHVI}} = \mathbb{E}\left[\mathrm{HVI}(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q)\right] = \int_{\mathbb{R}^M} \mathrm{HVI}(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q) d\boldsymbol{f} \tag{6}
$$

In the sequential setting and under the assumption of independent outcomes, qEHVI is simply EHVI and can be expressed in closed form [69]. However when $q > 1$, there is no known analytical formulation [70]. Instead, we estimate the expectation in (6) using MC integration with samples from the joint posterior $\mathbb{P}(\boldsymbol{f}(\boldsymbol{x}_1), \ldots, \boldsymbol{f}(\boldsymbol{x}_q)|\mathcal{D})$:

$$
\alpha_{q\mathrm{EHVI}} = \mathbb{E}\left[\mathrm{HVI}(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q)\right] \approx \frac{1}{N} \sum_{t=1}^N \mathrm{HVI}(\{\boldsymbol{f}_t(\boldsymbol{x}_i)\}_{i=1}^q) \tag{7}
$$

$$
= \frac{1}{N} \sum_{t=1}^N \sum_{k=1}^K \sum_{j=1}^q \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \prod_{m=1}^M \left[z_{k,X_j,t}^{(m)} - l_k^{(m)}\right]_+ \tag{8}
$$

where $\{\boldsymbol{f}_t(\boldsymbol{x}_i)\}_{i=1}^q \sim \mathbb{P}(\boldsymbol{f}(\boldsymbol{x}_1), \ldots, \boldsymbol{f}(\boldsymbol{x}_q)|X, Y)$ is the tth sample from the joint posterior over $\mathcal{X}_{\mathrm{cand}}$ and $\boldsymbol{z}_{k,X_j,t}^{(m)} = \min\left[\boldsymbol{u}_k, \min_{\boldsymbol{x}' \in X_j} \boldsymbol{f}_t(\boldsymbol{x}')\right]$.

A.3 Supporting Outcome Constraints

Recall that we defined the constrained hypervolume improvement as

$$
\mathrm{HVI}_{\mathrm{C}}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{c}(\boldsymbol{x})) = \mathrm{HVI}[\boldsymbol{f}(\boldsymbol{x})] \cdot \mathbb{1}[\boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}]. \tag{9}
$$

For $q = 1$ and assuming independence of the objectives and the constraints, the expected $\mathrm{HVI}_{\mathrm{C}}$ is the product of the expected HVI and the probability of feasibility (the expectation of $\mathbb{1}[\boldsymbol{c}(\boldsymbol{x}) \geq \boldsymbol{0}]$) [22]. However, requiring objectives and constraints to be independent is unnecessary when estimating the expectation with MC integration using samples from the joint posterior.

In the parallel setting, if all constraints are satisfied for all q candidates $\mathcal{X}_{\text{cand}} = \{\boldsymbol{x}_i\}_{i=1}^q$, HVI$_\text{C}$ is simply HVI. If a subset $\mathcal{V} \subset \mathcal{X}_{\text{cand}}, \mathcal{V} \neq \varnothing$ of the candidates violate at least one of the constraints, then the feasible HVI is the HVI of the set of feasible candidates: HVI$_\text{C}(\mathcal{X}_{\text{cand}}) = \text{HVI}(\mathcal{X}_{\text{cand}} \setminus \mathcal{V})$. That is, the *hypervolume contribution* (i.e. the marginal HVI) of an infeasible point is zero. In our formulation, HVI can be computed by multiplying (5) with an additional factor $\prod_{\boldsymbol{x}' \in X_j} \prod_{v=1}^V \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]$:

$$\text{HVI}_\text{C}(\{\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{c}(\boldsymbol{x}_i)\}_{i=1}^q) = \sum_{k=1}^K \sum_{j=1}^q \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \left[\left(\prod_{m=1}^M \left[z_{k,X_j}^{(m)} - l_k^{(m)} \right]_+ \right) \prod_{\boldsymbol{x}' \in X_j} \prod_{v=1}^V \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0] \right]. \tag{10}$$

The additional factor $\prod_{\boldsymbol{x}' \in X_j} \prod_{v=1}^V \mathbb{1}[c^{(v)}(\boldsymbol{x}_a) \geq 0]$ indicates whether all constraints are satisfied for all candidates in a given subset X_j. Thus HVI$_\text{C}$ can be computed in the same fashion as HVI, but with the additional step of setting the HV of all subsets containing $\boldsymbol{x}'$ to zero if $\boldsymbol{x}'$ violates any constraint. We can now again perform MC integration as in (5) to compute the expected constrained hypervolume improvement.

In this formulation, the marginal hypervolume improvement from a candidate is weighted by the probability that the candidate is feasible. The marginal hypervolume improvements are highly dependent on the outcomes of the other candidates. Importantly, the MC-based approach enables us to properly estimate the marginal hypervolume improvements across candidates by sampling from the joint posterior.

Note that while the *expected* constrained hypervolume $\mathbb{E}\left[\text{HVI}_\text{C}(\{\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{c}(\boldsymbol{x}_i)\}_{i=1}^q) \right]$ is differentiable, we may *not* differentiate inside the expectation (hence we cannot expect simply differentiating (10) on the sample-level to provide proper gradients). We therefore replace the indicator with a sigmoid function with temperature parameter ϵ, which provides a differentiable relaxation

$$\mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0] \approx s(c^{(v)}(\boldsymbol{x}'); \epsilon) := \frac{1}{1 + \exp(-c^{(v)}(\boldsymbol{x}')/\epsilon)} \tag{11}$$

that becomes exact in the limit $\epsilon \searrow 0$.

As in the unconstrained parallel scenario, there is no known analytical expression for the expected feasible hypervolume improvement. Therefore, we again use MC integration to approximate the expectation:

$$\alpha_{q\text{EHVI}_\text{C}}(\boldsymbol{x}) = \mathbb{E}\left[\text{HVI}_\text{C}(\{\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{c}(\boldsymbol{x}_i)\}_{i=1}^q) \right] \tag{12a}$$

$$\approx \frac{1}{N} \sum_{t=1}^N \text{HVI}_\text{C}(\{\boldsymbol{f}_t(\boldsymbol{x}_i), c_t(\boldsymbol{x}_i)\}_{i=1}^q) \tag{12b}$$

$$\approx \frac{1}{N} \sum_{t=1}^N \sum_{k=1}^K \sum_{j=1}^q \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \left[\left(\prod_{m=1}^M \left[z_{k,X_j,t}^{(m)} - l_k^{(m)} \right]_+ \right) \prod_{\boldsymbol{x}' \in X_j} \prod_{v=1}^V s(c^{(v)}(\boldsymbol{x}'); \epsilon) \right] \tag{12c}$$

A.3.1 Inclusion Exclusion principle for HVI$_\text{C}$

Equation (10) holds when the indicator function because HVI$_\text{C}$ is equivalent to HVI with the subset of feasible points. However, the sigmoid approximation can result in non-zero error. The error function $\varepsilon : 2^{\mathcal{X}_{\text{cand}}} \to \mathbb{R}$ can be expressed as

$$\varepsilon(X) = \prod_{\boldsymbol{x}' \in X} \prod_{v=1}^V \mathbb{1}[c(\boldsymbol{x}') > 0] - \prod_{\boldsymbol{x}' \in X} \prod_{v=1}^V s(c(\boldsymbol{x}'), \epsilon)$$

The error function gives a value to each to each element of $2^{\mathcal{X}_{\text{cand}}}$. Weight functions have been studied in conjunction with the inclusion-exclusion principle [56], but under the assumption of that the weight of a set is the sum of the weights of its elements: $w(A) = \sum_{a \in A} w(a)$. In our case, the weight function of a set A is the product the weights of its elements. There, it is not obvious whether the inclusion-exclusion principle will hold in this case.

Theorem 1. *Given a feasible Pareto front $\mathcal{P}_{feas}$, a partitioning $\{(\boldsymbol{l}_k, \boldsymbol{u}_k\}_{k=1}^K$ of the objective space $\mathbb{R}^M$ that is not dominated by the $\mathcal{P}_{feas}$, then for a set of points $\mathcal{X}_{cand}$ with objective values $\boldsymbol{f}(\mathcal{X}_{cand})$ and constraint values $\boldsymbol{c}(\mathcal{X}_{cand})$,*

$$\text{HVI}_\text{C}(\boldsymbol{f}(\mathcal{X}_{cand}), \boldsymbol{c}(\mathcal{X}_{cand}), \mathcal{P}, \boldsymbol{r}) = \text{HVI}(\boldsymbol{f}'(\mathcal{X}_{cand}), \mathcal{P}', \boldsymbol{r}')$$

where $\boldsymbol{f}'(\mathcal{X}_{cand})$ is the set of objective-constraint vectors for each candidate point $\boldsymbol{f}'(\boldsymbol{x}) \in \mathbb{R}^{M+V}$, $\mathcal{P}'$ is the set of vectors $[f^{(1)}(\boldsymbol{x}), ..., f^{(M)}(\boldsymbol{x}), \boldsymbol{0}_V] \in \mathbb{R}^{M+V}$, and $\boldsymbol{r}' = [r^{(1)}, ..., r^{(M)}, \boldsymbol{0}_V] \in \mathbb{R}^{M+V}$.

Proof. Recall equation 10,

$$\text{HVI}_\text{C}(\{\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{c}(\boldsymbol{x}_i)\}_{i=1}^q) = \sum_{k=1}^K \sum_{j=1}^q \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \left[\left(\prod_{m=1}^M \left[z_{k,X_j}^{(m)} - l_k^{(m)} \right]_+ \right) \prod_{\boldsymbol{x}' \in X_j} \prod_{v=1}^V \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0] \right].$$

Note that the constraint product

$$\prod_{\boldsymbol{x}' \in X_j} \prod_{v=1}^{V} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0] = \prod_{v=1}^{V} \prod_{\boldsymbol{x}' \in X_j} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]$$

$$= \prod_{v=1}^{V} \min_{\boldsymbol{x}' \in X_j} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]$$

$$= \prod_{v=1}^{V} \min\left[1, \min_{\boldsymbol{x}' \in X_j} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]\right] \qquad (13)$$

$$= \prod_{v=1}^{V} \left[\min\left[1, \min_{\boldsymbol{x}' \in X_j} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]\right] - 0\right].$$

For $v = 1, \ldots, V$, $k = 1, \ldots K$, let $l_k^{(M+v)} = 0$ and $u_k^{(M+v)} = 1$. Then, substituting into the following expression from Equation 13 gives

$$\min\left[1, \min_{\boldsymbol{x}' \in X_j} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]\right] = \min\left[u_k^{(M+v)}, \min_{\boldsymbol{x}' \in X_j} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]\right]$$

Recall from Section 4, that z is defined as: $\boldsymbol{z}_k := \min\left[\boldsymbol{u}_k, \boldsymbol{f}(\boldsymbol{x})\right]$. The high-level idea is that if we consider the indicator of the slack constraints $\mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]$ as objectives, then the above expression is consistent with the definition of z at the beginning of section 4. For $v = 1, \ldots, V$,

$$z_{k,X_j}^{(M+v)} = \min\left[1, \min_{\boldsymbol{x}' \in X_j} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]\right]$$

Thus,

$$\prod_{\boldsymbol{x}' \in X_j} \prod_{v=1}^{V} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0] = \prod_{v=1}^{V} \left[\min\left[1, \min_{\boldsymbol{x}' \in X_j} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]\right] - 0\right]$$

$$= \prod_{v=1}^{V} [z_{k,X_j}^{(M+v)} - l_k^{(M+v)}]_+$$

Returning to the $\mathrm{HVI_C}$ equation, we have

$$\mathrm{HVI_C}(\{\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{c}(\boldsymbol{x}_i)\}_{i=1}^{q}) = \sum_{k=1}^{K} \sum_{j=1}^{q} \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \left[\left(\prod_{m=1}^{M} [z_{k,X_j}^{(m)} - l_k^{(m)}]_+\right) \prod_{\boldsymbol{x}' \in X_j} \prod_{v=1}^{V} \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]\right]$$

$$= \sum_{k=1}^{K} \sum_{j=1}^{q} \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \left[\left(\prod_{m=1}^{M} [z_{k,X_j}^{(m)} - l_k^{(m)}]_+\right) \prod_{v=M+1}^{M+V} [z_{k,X_j}^{(v)} - l_k^{(M+v)}]_+\right]$$

$$= \sum_{k=1}^{K} \sum_{j=1}^{q} \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \left[\prod_{m=1}^{M+V} [z_{k,X_j}^{(m)} - l_k^{(m)}]_+\right]$$

$$(14)$$

$\square$

Now consider the case when a sigmoid approximation $\mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0] \approx s(c^{(v)}(\boldsymbol{x}'); \epsilon)$ is used. The only change to Equation 14 is that

$$z_{k,X_j}^{(m)} \approx \hat{z}_{k,X_j}^{(m)} = \min\left[u_k^{(M+v)}, \min_{\boldsymbol{x}' \in X_j} S[c^{(v)}(\boldsymbol{x}'), \epsilon]\right].$$

If $S[c^{(v)}(\boldsymbol{x}'), \epsilon] = \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]$ for all $v, \boldsymbol{x}'$, then HVI is computed exactly without approximation error. If $S[c^{(v)}(\boldsymbol{x}'), \epsilon] \mathbb{1}[c^{(v)}(\boldsymbol{x}') \geq 0]$ for any $v, \boldsymbol{x}'$, then there is approximation error: the hypervolume improvement from all subsets containing $\boldsymbol{x}'$ is proportional to $\prod_{v=1}^{V} \min_{\boldsymbol{x}' \in X} s(c(\boldsymbol{x}'), \epsilon)$. Since the constraint outcomes are directly considered as components in the hypervolume computation, the inclusion-exclusion principle incorporates the approximate indicator properly.

A.4 Complexity

Recall from Section 3.3 that, given posterior samples, the time complexity on a single-threaded machine is $T_1 = O(MNK(2^q - 1))$. The space complexity required for maximum parallelism is also is T_1 (ignoring the space required by the models), which does limit scalability to larger M and q, but difficulty scaling to large M is a known limitaiton of EHVI [69]. To reduce memory load, rectangles could be materialized and processed in chunks at the cost of additional runtime. In addition, our implementation of qEHVI uses the box decomposition algorithm from Couckuyt et al. [11], but we emphasize qEHVI is agnostic to the choice of partitioning algorithm and using a more efficient partitioning algorithm (e.g. [69, 17, 41]) may significantly improve memory footprint on GPU and enable larger using q in many scenarios.

B Error Bound on Sequential Greedy Approximation

If the acquisition function $\mathcal{L}(\mathcal{X}_{\text{cand}})$ is a normalized, monotone, submodular set function (where submodular means that the increase in $\mathcal{L}(\mathcal{X}_{\text{cand}})$ is non-increasing as elements are added to $\mathcal{X}_{\text{cand}}$ and normalized means that $\mathcal{L}(\emptyset) = 0$), then the sequential greedy approximation of $\mathcal{L}$ enjoys regret of no more than $\frac{1}{e}\mathcal{L}^*$, where $\mathcal{L}^*$ is the optima of $\mathcal{L}$ [23]. We have $\alpha_{q\text{EHVI}}(\mathcal{X}_{\text{cand}}) = \mathcal{L}(\mathcal{X}_{\text{cand}}) = \mathbb{E}_{\boldsymbol{f}}\big(\text{HVI}\big[\boldsymbol{f}(\mathcal{X}_{\text{cand}})\big]\big)$. Since HVI is a submodular set function [24] and the expectation of a stochastic submodular function is also submodular [2], $\alpha_{q\text{EHVI}}(\mathcal{X}_{\text{cand}})$ is also submodular and therefore its sequential greedy approximation enjoys regret of no more than $\frac{1}{e}\alpha_{q\text{EHVI}}^*$. Using the result from Wilson et al. [65], the MC-based approximation $\hat{\alpha}_{q\text{EHVI}}(\mathcal{X}_{\text{cand}}) = \sum_{t=1}^{N} \text{HVI}\big[\boldsymbol{f}_t(\mathcal{X}_{\text{cand}})\big]$ also enjoys the same regret bound since HVI is a normalized submodular set function.[7]

C Convergence Results

For the purpose of stating our convergence results, we recall some concepts and notation from Balandat et al. [5]. First, consider a sample $\{\boldsymbol{f}_t(\boldsymbol{x}_1)\}_{i=1}^{q}$ from the multi-output posterior of the GP surrogate model. Let $\boldsymbol{x} \in \mathbb{R}^{qd}$ be the stacked set of candidates $\mathcal{X}_{\text{cand}}$ and let $\boldsymbol{f}_t(\boldsymbol{x}) := [f_t(\boldsymbol{x}_1)^T, \ldots, f_t(\boldsymbol{x}_q)^T]^T$ be the stacked set of corresponding objective vectors. It is well known that, using the reparameterization trick, we can write

$$\boldsymbol{f}_t(\boldsymbol{x}) = \mu(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon_t, \tag{15}$$

where $\mu : \mathbb{R}^{qd} \to \mathbb{R}^{qM}$ is the mean function of the multi-output GP, $L(\boldsymbol{x}) \in \mathbb{R}^{qM \times qM}$ is a root decomposition (typically the Cholesky decomposition) of the multi-output GP's posterior covariance $\Sigma(\boldsymbol{x}) \in \mathbb{R}^{qM \times qM}$, and $\epsilon_t \in \mathbb{R}^{qM}$ with $\epsilon_t \sim \mathcal{N}(0, I_{qM})$.

For $\boldsymbol{x} \in \mathcal{X}$, consider the MC-approximation $\hat{\alpha}_{q\text{EHVI}}^N(\boldsymbol{x})$ from (5). Denote by $\nabla_{\boldsymbol{x}}\hat{\alpha}_{q\text{EHVI}}^N(\boldsymbol{x})$ the gradient of $\hat{\alpha}_{q\text{EHVI}}^N(\boldsymbol{x})$, obtained by averaging the gradients on the sample-level:

$$\nabla_{\boldsymbol{x}}\hat{\alpha}_{q\text{EHVI}}^N(\boldsymbol{x}) := \frac{1}{N}\sum_{t=1}^{N}\nabla_{\boldsymbol{x}}\text{HVI}(\{f_t(\boldsymbol{x}_i)\}_{i=1}^{q}) \tag{16}$$

Let $\alpha_{q\text{EHVI}}^* := \max_{\boldsymbol{x} \in \mathcal{X}} \alpha_{q\text{EHVI}}(\boldsymbol{x})$ denote the maximum of the true acquisition function qEHVI, and let $\mathcal{X}^* := \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha_{q\text{EHVI}}(\boldsymbol{x})$ denote the set of associated maximizers.

Theorem 2. *Suppose that $\mathcal{X}$ is compact and that f has a Multi-Output Gaussian Process prior with continuously differentiable mean and covariance functions. If the base samples $\{\epsilon_t\}_{t=1}^N$ are drawn i.i.d. from $\mathcal{N}(0, I_{qM})$, and if $\hat{\boldsymbol{x}}_N^* \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} \hat{\alpha}_{q\text{EHVI}}^N(\boldsymbol{x})$, then*

(1) $\alpha_{q\text{EHVI}}(\hat{\boldsymbol{x}}_N^*) \to \alpha_{q\text{EHVI}}^*$ a.s.

(2) $\text{dist}(\hat{\boldsymbol{x}}_N^*, \mathcal{X}^*) \to 0$ a.s.

In addition to the almost sure convergence in Theorem 2, deriving a result on the convergence rate of the optimizer, similar to the one obtained in [5], should be possible. We leave this to future work. Moreover, the results in Theorem 2 can also be extended to the situation in which the base samples are generated using a particular class of randomized QMC methods (see similar results in [5]).

Proof. We consider the setting from Balandat et al. [5, Section D.5]. Let $\epsilon \sim \mathcal{N}(0, I_{qM})$, so that we can write the posterior over outcome m at $\boldsymbol{x}$ as the random variable $f^{(m)}(\boldsymbol{x}, \epsilon) = S_{\{i_j, m\}}(\mu(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon)$, where $\mu(\boldsymbol{x})$

---

[7]As noted in Wilson et al. [65], submodularity technically requires the search space $\mathcal{X}$ to be finite, whereas in BO, it will typically be infinite. Wilson et al. [65] note that in similar scenarios, submodularity has been extended to infinite sets $\mathcal{X}$ (e.g. Srinivas et al. [58]).

and $L(\boldsymbol{x})$ are the (vector-valued) posterior mean and the Cholesky factor of posterior covariance, respectively, and $S_{\{i_j, m\}}$ is an appropriate selection matrix (in particular, $\|S_{\{i_j,m\}}\|_\infty \leq 1$ for all i_j and m). Let

$$A(\boldsymbol{x}, \epsilon) = \sum_{k=1}^{K} \sum_{j=1}^{q} \sum_{X_j \in \mathcal{X}_j} (-1)^{j+1} \prod_{m=1}^{M} \left[z_{k,X_j}^{(m)}(\epsilon) - l_k^{(m)}\right]_+$$

where

$$z_{k,X_j}^{(m)}(\epsilon) = \min\left[u_k^{(m)}, f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots, f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right]$$

and $X_j = \{\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_j}\}$. Following [5, Theorem 3], we need to show that there exists an integrable function $\ell : \mathbb{R}^{q \times M} \mapsto \mathbb{R}$ such that for almost every ϵ and all $\boldsymbol{x}, \boldsymbol{y} \subseteq \mathcal{X}, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{q \times d}$,

$$|A(\boldsymbol{x}, \epsilon) - A(\boldsymbol{y}, \epsilon)| \leq \ell(\epsilon)\|\boldsymbol{x} - \boldsymbol{y}\|. \tag{17}$$

Let us define

$$\tilde{a}_{kmjX_j}(\boldsymbol{x}, \epsilon) := \left[\min\left[u_k^{(m)}, f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots, f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right] - l_k^{(m)}\right]_+.$$

Linearity implies that it suffices to show that this condition holds for

$$\tilde{A}(\boldsymbol{x}, \epsilon) := \prod_{m=1}^{M} \tilde{a}_{kmjX_j}(\boldsymbol{x}, \epsilon) = \prod_{m=1}^{M} \left[\min\left[u_k^{(m)}, f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots, f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right] - l_k^{(m)}\right]_+ \tag{18}$$

for all k, j, and X_j. Observe that

$$\tilde{a}_{kmjX_j}(\boldsymbol{x}, \epsilon) \leq \left|\min\left[u_k^{(m)}, f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots, f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right] - l_k^{(m)}\right|$$

$$\leq |l_k^{(m)}| + \left|\min\left[u_k^{(m)}, f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots, f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right]\right|.$$

Note that if $u_k^{(m)} = \infty$, then $\min[u_k^{(m)}, f(\boldsymbol{x}, \epsilon)_{i_1}^{(m)}, \ldots f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)] = \min[f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)]$. If $u_k^{(m)} < \infty$, then $\min[u_k^{(m)}, f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)] < \left|\min[f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)]\right| + |u_k^{(m)}|$. Let $w_k^{(m)} = u_k^{(m)}$ if $u_k^{(m)} < \infty$ and 0 otherwise. Then

$$\tilde{a}_{kmjX_j}(\boldsymbol{x}, \epsilon) \leq |l_k^{(m)}| + |w_k^{(m)}| + \left|\min\left[f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots, f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right]\right|$$

$$\leq |l_k^{(m)}| + |w_k^{(m)}| + \sum_{i_1, \ldots, i_j} \left|f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right|.$$

We therefore have that

$$\left|\tilde{a}_{kmjX_j}(\boldsymbol{x}, \epsilon)\right| \leq |l_k^{(m)}| + |w_k^{(m)}| + |X_j|\left(\|\mu^{(m)}(\boldsymbol{x})\| + \|L^{(m)}(\boldsymbol{x})\|\|\epsilon\|\right)$$

for all k, m, j, X_j, where $|X_j|$ denotes the cardinality of the set X_j. Under our assumptions (compactness of $\mathcal{X}$, continuous differentiability of mean and covariance function), both $\mu(\boldsymbol{x})$ and $L(\boldsymbol{x})$, as well as their respective gradients w.r.t. $\boldsymbol{x}$, are uniformly bounded. In particular there exist $C_1, C_2 < \infty$ such that

$$\left|\tilde{a}_{kmjX_j}(\boldsymbol{x}, \epsilon)\right| \leq C_1 + C_2\|\epsilon\|$$

for all k, m, j, X_j.

Dropping indices k, j, X_j for simplicity, observe that

$$\left|\tilde{A}(\boldsymbol{x}, \epsilon) - \tilde{A}(\boldsymbol{y}, \epsilon)\right| = \left|\tilde{a}_1(\boldsymbol{x}, \epsilon)\tilde{a}_2(\boldsymbol{x}, \epsilon) - \tilde{a}_1(\boldsymbol{y}, \epsilon)\tilde{a}_2(\boldsymbol{y}, \epsilon)\right| \tag{19a}$$

$$= \left|\tilde{a}_1(\boldsymbol{x}, \epsilon)\big(\tilde{a}_2(\boldsymbol{x}, \epsilon) - \tilde{a}_2(\boldsymbol{y}, \epsilon)\big) + \tilde{a}_2(\boldsymbol{y}, \epsilon)\big(\tilde{a}_1(\boldsymbol{x}, \epsilon) - \tilde{a}_1(\boldsymbol{y}, \epsilon)\big)\right| \tag{19b}$$

$$\leq \left|\tilde{a}_1(\boldsymbol{x}, \epsilon)\right|\left|\tilde{a}_2(\boldsymbol{x}, \epsilon) - \tilde{a}_2(\boldsymbol{y}, \epsilon)\right| + \left|\tilde{a}_2(\boldsymbol{y}, \epsilon)\right|\left|\tilde{a}_1(\boldsymbol{x}, \epsilon) - \tilde{a}_1(\boldsymbol{y}, \epsilon)\right|. \tag{19c}$$

Furthermore,

$$\left|\tilde{a}_{kmjX_j}(\boldsymbol{x}, \epsilon) - \tilde{a}_{kmjX_j}(\boldsymbol{y}, \epsilon)\right| \leq \sum_{i_1, \ldots, i_j} \left|S_{\{i_j, m\}}(\mu(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon) - S_{\{i_j, m\}}(\mu(\boldsymbol{y}) + L(\boldsymbol{y})\epsilon)\right|$$

$$\leq |X_j|\left(\|\mu(\boldsymbol{x}) - \mu(\boldsymbol{y})\| + \|L(\boldsymbol{x}) - L(\boldsymbol{y})\|\|\epsilon\|\right).$$

Since μ and L have uniformly bounded gradients, they are Lipschitz. Therefore, there exist $C_3, C_4 < \infty$ such that

$$\left|\tilde{a}_{kmjX_j}(\boldsymbol{x}, \epsilon) - \tilde{a}_{kmjX_j}(\boldsymbol{y}, \epsilon)\right| \leq (C_3 + C_4\|\epsilon\|)\|\boldsymbol{x} - \boldsymbol{y}\|$$

20

for all $\boldsymbol{x}, \boldsymbol{y}, k, m, j, X_j$. Plugging this into (19) above, we find that

$$\left|\tilde{A}(\boldsymbol{x}, \epsilon) - \tilde{A}(\boldsymbol{y}, \epsilon)\right| \leq 2\Big(C_1 C_3 + (C_1 C_4 + C_2 C_3)\|\epsilon\| + C_2 C_4 \|\epsilon\|^2\Big)\|\boldsymbol{x} - \boldsymbol{y}\|$$

for all $\boldsymbol{x}, \boldsymbol{y}$ and ϵ. For $M > 2$ we generalize the idea from (19), making sure to telescope the respective expressions. It is not hard to see that with this, there exist $C < \infty$ such that

$$\left|\tilde{A}(\boldsymbol{x}, \epsilon) - \tilde{A}(\boldsymbol{y}, \epsilon)\right| \leq C \sum_{m=1}^{M} \|\epsilon\|^m \|\boldsymbol{x} - \boldsymbol{y}\|$$

Letting $\ell(\epsilon) := C \sum_{m=1}^{M} \|\epsilon\|^m$, we observe that $\ell(\epsilon)$ is integrable (since all absolute moments exist for the Normal distribution).

The result now follows from in Balandat et al. [5, Theorem 3]. $\qquad\square$

Besides the above convergence result, we can also show that the sample average gradient of the MC approximation of qEHVI is an unbiased estimator of the true gradient of qEHVI:

Proposition 1. *Suppose that the GP mean and covariance function are continuously differentiable. Suppose further that the candidate set $\boldsymbol{x}$ has no duplicates, and that the sample-level gradients $\nabla_{\boldsymbol{x}} \mathrm{HVI}(\{f_t(\boldsymbol{x}_i)\}_{i=1}^q)$ are obtained using the reparameterization trick as in [5]. Then*

$$\mathbb{E}\big[\nabla_{\boldsymbol{x}} \hat{\alpha}_{q\mathrm{EHVI}}^N(\boldsymbol{x})\big] = \nabla_{\boldsymbol{x}} \alpha_{q\mathrm{EHVI}}(\boldsymbol{x}), \tag{20}$$

that is, the averaged sample-level gradient is an unbiased estimate of the gradient of the true acquisition function.

Proof. This proof follows the arguments Wang et al. [63, Theorem 1], which leverages Glasserman [31, Theorem 1]. We verify the conditions of Glasserman [31, Theorem 1] below. Using the arguments from [5], we know that, under the assumption of differentiable mean and covariance functions, the samples $\boldsymbol{f}_t(\boldsymbol{x})$ are continuously differentiable w.r.t. $\boldsymbol{x}$ (since there are no duplicates, and thus the covariance $\Sigma(\boldsymbol{x})$ is non-singular). Hence, Glasserman [31, A1] is satisfied. Furthermore, it is easy to see from (1) that $\mathrm{HVI}(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q)$ is *a.s.* continuous and is differentiable w.r.t. $\boldsymbol{f}_t(\boldsymbol{x})$ on $\mathbb{R}^M$, except on the edges of the hyper-rectangle decomposition $\{S_k\}_{k=1}^K$ of the non-dominated space, which satisfies [31, A3]. The set of points defined by the union of these edges clearly has measure zero under any non-degenerate (non-singular covariance) GP posterior on $\mathbb{R}^M$, so Glasserman [31, A4] holds. Therefore Glasserman [31, Lemma 2] holds, so $\mathrm{HVI}(\{\boldsymbol{f}(\boldsymbol{x}_i)\}_{i=1}^q)$ is *a.s.* piece-wise differentiable w.r.t. $\boldsymbol{x}$.

Lastly, we need to show that the result in Glasserman [31, Lemma 3] holds:

$$\mathbb{E}\bigg[\sup_{x_{ci} \notin \tilde{D}} |A'(\boldsymbol{x}, \epsilon)|\bigg] < \infty.$$

As in Wang et al. [63, Theorem 1], we fix $\boldsymbol{x}$ except for x_{ci} where x_{ci} is the cth component of the ith point, We need to show that $\mathbb{E}\big[\sup_{x_{ci} \notin \tilde{D}} |A'(\boldsymbol{x}, \epsilon)|\big] < \infty$. By linearity, it suffices to show that $\mathbb{E}\big[\sup_{x_{ci} \notin \tilde{D}} |\tilde{A}'(\boldsymbol{x}, \epsilon)|\big] < \infty$. We have

$$\mathbb{E}\bigg[\sup_{x_{ci} \notin \tilde{D}} |\tilde{A}'(\boldsymbol{x}, \epsilon)|\bigg] = \mathbb{E}\bigg[\sup_{x_{ci} \notin \tilde{D}} \left|\frac{\partial \tilde{A}(\boldsymbol{x}, \epsilon)}{\partial x_{ci}}\right|\bigg].$$

Consider the $M = 2$ case. We have $\tilde{A}(\boldsymbol{x}, \epsilon) = a_1(\boldsymbol{x}, \epsilon) a_2(\boldsymbol{x}, \epsilon)$, where

$$a_m(\boldsymbol{x}, \epsilon) = \left[\min\left[u_k^{(m)}, f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots, f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right] - l_k^{(m)}\right]_+.$$

The partial derivative of $\tilde{A}(\boldsymbol{x}, \epsilon)$ with respect to x_{ci} is

$$\frac{\partial \tilde{A}(\boldsymbol{x}, \epsilon)}{\partial x_{ci}} = \frac{\partial a_1(\boldsymbol{x}, \epsilon)}{\partial x_{ci}} a_2(\boldsymbol{x}, \epsilon) + a_1(\boldsymbol{x}, \epsilon) \frac{\partial a_2(\boldsymbol{x}, \epsilon)}{\partial x_{ci}},$$

and therefore

$$\left|\frac{\partial \tilde{A}(\boldsymbol{x}, \epsilon)}{\partial x_{ci}}\right| \leq \left|\frac{\partial a_1(\boldsymbol{x}, \epsilon)}{\partial x_{ci}}\right| \cdot \left|a_2(\boldsymbol{x}, \epsilon)\right| + \left|a_1(\boldsymbol{x}, \epsilon)\right| \cdot \left|\frac{\partial a_2(\boldsymbol{x}, \epsilon)}{\partial x_{ci}}\right|$$

Since we are only concerned with $x_{ci} \notin \tilde{D}$,

$$a_m(\boldsymbol{x}, \epsilon) = \left[\min\left[f^{(m)}(\boldsymbol{x}_{i_1}, \epsilon), \ldots, f^{(m)}(\boldsymbol{x}_{i_j}, \epsilon)\right] - l_k^{(1)}\right]_+.$$

As in the proof of Theorem 2, we write the posterior over outcome m at $\boldsymbol{x}$ as the random variable $f^{(m)}(\boldsymbol{x}, \epsilon) = S_{\{i_j,m\}}(\boldsymbol{\mu}(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon)$, where $\epsilon \sim \mathcal{N}(0, I_{qM})$ and $S_{\{i_j,m\}}$ is an appropriate selection matrix. With this,

$$a_m(\boldsymbol{x}, \epsilon) = \left[\min\left[S_{\{i_1,1\}}(\mu(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon), \ldots, S_{\{i_j,1\}}(\mu(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon)\right] - l_k^{(1)}\right]_+.$$

Since the interval $\mathcal{X}$ is compact and the mean, covariance, and Cholesky factor of the covariance $\mu(\boldsymbol{x}), C(\boldsymbol{x}), L(\boldsymbol{x})$ are continuously differentiable, for all m we have

$$\sup_{x_{ci}}\left|\frac{\partial \mu^{(m)}(\boldsymbol{x}_a)}{\partial x_{ci}}\right| = \mu_a^{*,(m)} < \infty, \qquad \sup_{x_{ci}}\left|\frac{\partial L^{(m)}(\boldsymbol{x})}{\partial x_{ci}}\right| = L_{ca}^{*,(m)} < \infty.$$

Let $\mu_{**}^{(m)} = \max_a \mu_a^{*,(m)}$, $L_{**}^{(m)} = \max_{a,b} L_{ab}^{*,(m)}(\boldsymbol{x})$, where $L_{ab}^{(m)}$ is the element at row a, column b in $L^{(m)}$, the Cholesky factor for outcome m. Let $\epsilon^{(m)} \in \mathbb{R}^q$ denote the vector of i.i.d. $\mathcal{N}(0, 1)$ samples corresponding to outcome m. Then we have

$$\left|\frac{\partial}{\partial x_{ci}}\left[\left[\min\left[S_{\{i_1,1\}}(\mu(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon), \ldots, S_{\{i_j,1\}}(\mu(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon)\right] - l_k^{(1)}\right]_+\right|$$

$$\leq \left|\left[\mu_{**}^{(m)} + L_{**}^{(m)}||\epsilon^{(m)}||_1 - l_k^{(m)}\right]_+\right|$$

$$\leq \left|\mu_{**}^{(m)} + L_{**}^{(m)}||\epsilon^{(m)}||_1\right| + \left|l_k^{(m)}\right|.$$

Under our assumptions (compactness of $\mathcal{X}$, continuous differentiability of mean and covariance function) both $\boldsymbol{\mu}(\boldsymbol{x})$ and $L(\boldsymbol{x})$, as well as their respective gradients, are uniformly bounded. In particular there exist $C_1^{(m)}, C_2^{(m)} < \infty$ such that

$$\left|S_{\{a,m\}}(\mu(\boldsymbol{x}) + L(\boldsymbol{x})\epsilon) - l_k^{(m)}\right| \leq C_1^{(m)} + C_2^{(m)}||\epsilon^{(m)}||_1$$

for all $a = i_1, \ldots, i_j$.

Hence,

$$\left|\frac{\partial \tilde{A}(\boldsymbol{x}, \epsilon)}{\partial x_{ci}}\right| \leq \left[\left|\mu_{**}^{(1)} + C_{**}^{(1)}||\epsilon^{(1)}||_1\right| + \left|l_k^{(1)}\right|\right]\left[C_1^{(2)} + C_2^{(2)}||\epsilon^{(2)}||_1\right]$$

$$+ \left[C_1^{(1)} + C_2^{(1)}||\epsilon^{(1)}||_1\right]\left[\left|\mu_{**}^{(2)} + C_{**}^{(2)}||\epsilon^{(2)}||_1\right| + \left|l_k^{(2)}\right|\right]$$

Since ϵ is absolutely integrable,

$$\mathbb{E}\left(\left|\frac{\partial \tilde{A}(\boldsymbol{x}, \epsilon)}{\partial x_{ci}}\right|\right) < \infty.$$

Hence, $\mathbb{E}\left[\sup_{x_{ci} \notin \tilde{D}}|A'(\boldsymbol{x}, \epsilon)|\right] < \infty$. This can be extended to $M > 2$ in the same manner using the product rule to obtain

$$\mathbb{E}\left(\frac{\partial \tilde{A}(\boldsymbol{x}, \epsilon)}{\partial x_{ci}}\right) \leq \sum_{m=1}^{M}\left(\left[\left|\mu_{**}^{(m)} + C_{**}^{(m)}\mathbb{E}[||\epsilon^{(m)}||_1]\right| + \left|l_k^{(1)}\right|\right]\prod_{n=1, n\neq m}^{M}\left[C_1^{(n)} + C_2^{(n)}\mathbb{E}[||\epsilon^{(n)}||_1]\right]\right)$$

$$\leq \sum_{m=1}^{M}\left(\left[\left|\mu_{**}^{(m)} + \frac{\pi}{2}qC_{**}^{(m)}\right| + \left|l_k^{(1)}\right|\right]\prod_{n=1, n\neq m}^{M}\left[C_1^{(n)} + \frac{\pi}{2}qC_2^{(n)}\right]\right).$$

Hence, $\mathbb{E}\left[\sup_{x_{ci} \notin \tilde{D}}|A'(\boldsymbol{x}, \epsilon)|\right] < \infty$ for $M \geq 2$ and Glasserman [31, Theorem 1] holds. $\qquad\square$

D Monte-Carlo Approximation

Figure 5b shows the gradient of analytic EHVI and the MC estimator qEHVI on slice of a 3-objective problem. Even using only $N = 32$ QMC samples, the average sample gradient has very low variance. Moreover, fixing the base samples also greatly reduces the variance without introducing bias.
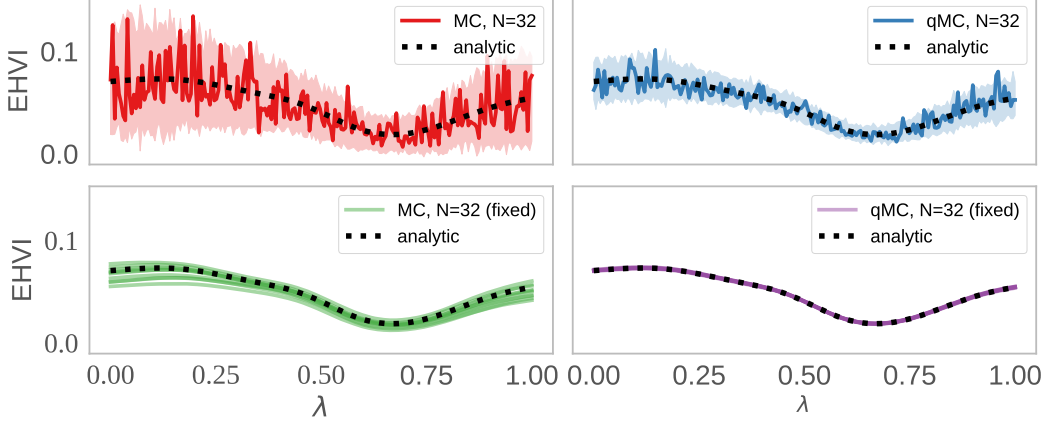
(a) A comparison of the analytic EHVI acquisition function and the MC-based qEHVI for $q = 1$.
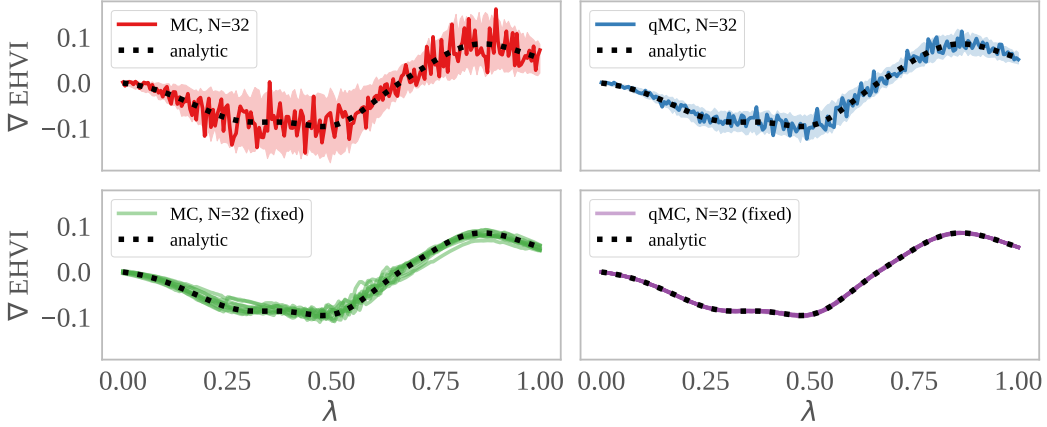
(b) A comparison of the exact gradient of analytic EHVI and the exact sample average gradient of the MC-based qEHVI for $q = 1$.

Figure 5: A comparison of (a) the analytic EHVI and MC-based qEHVI for $q = 1$ and (b) a comparison of the exact gradient $\nabla\alpha_{\mathrm{EHVI}}$ of analytic EHVI and average sample gradient of the MC-estimator $\nabla\hat{\alpha}_{q\mathrm{EHVI}}$ over a slice of the input space on a DTLZ2 problem ($q = 1$, $M = 3$, $d = 6$) [15]. $x^{(0)}$ is varied across $0 \leq \lambda \leq 1$, while $x^{(i)}$ for $1, ...D$ are held constant. In each of (a) and (b), the top row show qEHVI where the (quasi-)standard normal base samples are resampled for each value of $x^{(0)}$. The solid line is one sample average (across (q)MC samples) and the shaded area is the mean plus 2 standard errors across 50 repetitions. The bottom row uses the same base samples for evaluating each test point and the sample average for each of 50 repetitions is plotted.

E Experiment Details

E.1 Algorithms

For TS-TCH, we draw a sample from the joint posterior over a discrete set of $1000d$ points sampled from a scrambled Sobol sequence. For PESMO, we follow [27] and use a Pareto set of size 10 for each sampled GP, which is optimized over a discrete set of $1000d$ points sampled from a scrambled Sobol sequence. The current

Table 2: Reference points for all benchmark problems. Assuming minimization. In our benchmarks, equivalently maximize the negative objectives and multiply the reference points by -1.

PROBLEM	REFERENCE POINT
BRANINCURRIN	(18.0, 6.0)
DTLZ2	$(1.1, ..., 1.1) \in \mathbb{R}^M$
ABR	(-150.0, 3500.0, 5.1)
VEHICLE CRASH SAFETY	(1864.72022, 11.81993945, 0.2903999384)
CONSTRAINEDBRANINCURRIN	(90.0, 10.0)
C2-DTLZ2	$(1.1, ..., 1.1) \in \mathbb{R}^M$

Pareto front is approximated by optimizing the posterior means over a grid as is done in Garrido-Merchán and Hernández-Lobato [26, 27]. For SMS-EGO, we use the observed Pareto front. All acquisition functions are optimized with L-BFGS-B (with a maximum of 200 iterations); SMS-EGO [53] and PESMO [26] use gradients approximated by finite differences and all other methods use exact gradients. For all methods, each outcome is modeled with an independent Gaussian process with a Matern $5/2$ ARD kernel. The methods implemented in Spearmint use a fully Bayesian treatment of the hyperparameters with 10 samples from posterior over the hyperparamters, and the methods implemented in BoTorch use maximum a posteriori estimates of the GP hyperparameters. All methods are initialized with $2(d + 1)$ points from a scrambled Sobol sequence. qPAREGO and qEHVI use $N = 128$ QMC samples.

E.1.1 Reference point specification

There is a large body of literature on the effects of reference point specification [4, 35, 36]. The hypervolume indicator is sensitive to specified the reference point: a reference point that is far away from the Pareto front will favor extreme points, where as reference point that is close to the Pareto front gives more weight to less extreme points [36]. Sensitivity to the reference point is affects both the evaluation of different MO methods and the utility function for methods that rely HV. In practice, a decision maker may be able to specify a reference point that satisfies their preference with domain knowledge. If a reference point is provided by the decision maker, previous work has suggested heuristics for choosing reference points for use in an algorithm's utility function [35, 53]. We follow previous work [69, 68] and assume that the reference point is known.

We also considered (but did not use in our experiments) a dynamic reference point strategy where at each BO iteration, the reference point is selected to be a point slightly worse than the nadir (component-wise minimum) point of the current observed Pareto front for computing the acquisition function: $\boldsymbol{r} = \boldsymbol{y}_{\text{nadir}} - 0.1 \cdot |\boldsymbol{y}_{\text{nadir}}|$ where $\boldsymbol{y}_{\text{nadir}} = \left(\min_{y^{(1)} \in \mathcal{D}^{(1)}} y^{(1)}, \ldots, \min_{y^{(m)} \in \mathcal{D}^{(m)}} y^{(m)} \right)$. This reference point is used in SMS-EMOA in Ishibuchi et al. [35]), and we find similar average performance (but higher variance) on problems to using a known reference point with continuous Pareto fronts. If the Pareto front is discontinuous, then it is possible not all sections of the Pareto front will be reached.

E.1.2 qPAREGO

Previous work has only considered unconstrained sequential optimization with ParEGO [40, 7] and ParEGO is often optimized with gradient-free methods [53]. To the best of our knowledge, qPAREGO is the first to support parallel and constrained optimization. Moreover, we compute exact gradients via auto-differentiation for acquisition optimization. ParEGO is typically implemented by applying augmented Chebyshev scalarization and modeling the scalarized outcome [40]. However, recent work has shown that composite objectives offer improved optimization performance [3]. qPAREGO uses a MC-based Expected Improvement [38] acquisition function, where the objectives are modeled independently and the augmented Chebyshev scalarization [40] is applied to the posterior samples as a composite objective. This approach enables the use of sequential greedy optimization of q candidates with proper integration over the posterior at the pending points. Importantly, the sequential greedy approach allows for using different random scalarization weights for selecting each of the q candidates. qPAREGO is extended to the constrained setting by weighting the EI by the probability of feasibility [25]. We estimate the probability of feasiblity using the posterior samples and approximate the indicator function with a sigmoid to maintain differentiablity as in constrained qEHVI. qPAREGO is trivially extended to the noisy setting using Noisy Expected Improvement [43, 5], but we use Expected Improvement in our experiments as all of the problems are noiseless.

E.2 Benchmark Problems

The details for the benchmark problems below assume minimization of all objectives. Table 2 provides the reference points used for all benchmark problems.

Branin-Currin

$$f^{(1)}(x_1', x_2') = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - r)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$$

$$f^{(2)}(x_1, x_2) = \left[1 - \exp\left(-\frac{1}{(2x_2)}\right)\right]\frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}$$

where $x_1, x_2 \in [0, 1]$, $x_1' = 15x_1 - 5$, and $x_2' = 15x_2$.

The constrained Branin-Currin problem uses the following disk constraint from [29]:

$$c(x_1', x_2') = 50 - (x_1' - 2.5)^2 - (x_2' - 7.5)^2) \geq 0$$

DTLZ2 The objectives are given by [15]:

$$f_1(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_M))\cos\left(\frac{\pi}{2}x_1\right)\cdots\cos\left(\frac{\pi}{2}x_{M-2}\right)\cos\left(\frac{\pi}{2}x_{M-1}\right)$$

$$f_2(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_M))\cos\left(\frac{\pi}{2}x_1\right)\cdots\cos\left(\frac{\pi}{2}x_{M-2}\right)\sin\left(\frac{\pi}{2}x_{M-1}\right)$$

$$f_3(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_M))\cos\left(\frac{\pi}{2}x_1\right)\cdots\sin\left(\frac{\pi}{2}x_{M-2}\right)$$

$$\vdots$$

$$f_M(\boldsymbol{x}) = (1 + g(\boldsymbol{x}_M))\sin\left(\frac{\pi}{2}x_1\right)$$

where $g(\boldsymbol{x}) = \sum_{x_i \in \boldsymbol{x}_M}(x_i - 0.5)^2$, $\boldsymbol{x} \in [0, 1]^d$, and $\boldsymbol{x}_M$ represents the last $d - M + 1$ elements of $\boldsymbol{x}$.

The C2-DTLZ2 problem adds the following constraint [16]:

$$c(\boldsymbol{x}) = -\min\left[\min_{i=1}^{M}\left((f_i(\boldsymbol{x}) - 1)^2 + \sum_{j=1, j=i}^{M}(f_j^2 - r^2)\right), \left(\sum_{i=1}^{M}\left((f_i(\boldsymbol{x}) - \frac{1}{\sqrt{M}})^2 - r^2\right)\right)\right] \geq 0$$

Vehicle Crash Safety The objectives are given by [60]:

$$f_1(\boldsymbol{x}) = 1640.2823 + 2.3573285x_1 + 2.3220035x_2 + 4.5688768x_3 + 7.7213633x_4 + 4.4559504x_5$$

$$f_2(\boldsymbol{x}) = 6.5856 + 1.15x_1 - 1.0427x_2 + 0.9738x_3 + 0.8364x_4 - 0.3695x_1x_4 + 0.0861x_1x_5$$
$$+ 0.3628x_2x_4 + 0.1106x_1^2 - 0.3437x_3^2 + 0.1764x_4^2$$

$$f_3(\boldsymbol{x}) = -0.0551 + 0.0181x_1 + 0.1024x_2 + 0.0421x_3 - 0.0073x_1x_2 + 0.024x_2x_3 - 0.0118x_2x_4$$
$$- 0.0204x_3x_4 - 0.008x_3x_5 - 0.0241x_2^2 + 0.0109x_4^2$$

where $\boldsymbol{x} \in [1, 3]^5$.

Policy Optimization for Adaptive Bitrate Control The controller is given by: $a_t = x_0\hat{z}_{\text{bd},t} + x_2 z_{\text{bf},t} + x_3$, where $\hat{z}_{\text{bd},t} = \frac{\sum_{t_i < t} z_{\text{bd},t_i}\exp(-x_1 t_i)}{\sum_{t_i < t}\exp(-x_1 t_i)}$ is estimated bandwidth at time t using an exponential moving average, $z_{\text{bf},t}$ is the buffer occupancy at time t, and $x_0, ...x_3$ are the parameters we seek to optimize. We evaluate each policy on a set of 400 videos, where the number of time steps (chunks) in each video stream trajectory depends on the size of the video.

Table 3: Acquisition Optimization wall time in seconds on a CPU (2x Intel Xeon E5-2680 v4 @ 2.40GHz) and on a GPU (Tesla V100-SXM2-16GB). The mean and two standard errors are reported. NA indicates that the algorithm does not support constraints.

CPU	CONSTRAINEDBRANINCURRIN	DTLZ2
PESMO (q=1)	NA	278.53 ($\pm$25.66)
SMS-EGO (q=1)	NA	104.26 ($\pm$7.66)
TS-TCH (q=1)	NA	52.55 ($\pm$0.06)
qPAREGO (q=1)	2.4 ($\pm$0.37)	4.68 ($\pm$0.46)
EHVI (q=1)	NA	3.58 ($\pm$0.28)
qEHVI (q=1)	5.69 ($\pm$0.43)	5.95 ($\pm$0.45)
GPU	CONSTRAINEDBRANINCURRIN	DTLZ2
TS-TCH (q=1)	NA	0.25 ($\pm$0.00)
TS-TCH (q=2)	NA	0.27 ($\pm$0.00)
TS-TCH (q=4)	NA	0.28 ($\pm$0.00)
TS-TCH (q=8)	NA	0.32 ($\pm$0.01)
qPAREGO (q=1)	3.52 ($\pm$0.34)	9.04 ($\pm$0.93)
qPAREGO (q=2)	6.0 ($\pm$0.56)	14.23 ($\pm$1.55)
qPAREGO (q=4)	12.07 ($\pm$0.98)	40.5 ($\pm$3.21)
qPAREGO (q=8)	33.1 ($\pm$3.32)	84.15 ($\pm$6.9)
EHVI (q=1)	NA	84.15 ($\pm$6.9)
qEHVI (q=1)	5.61 ($\pm$0.17)	10.21 ($\pm$0.58)
qEHVI (q=2)	19.06 ($\pm$5.88)	17.75 ($\pm$0.97)
qEHVI (q=4)	29.26 ($\pm$2.01)	40.41 ($\pm$2.78)
qEHVI (q=8)	91.56 ($\pm$5.51)	106.51 ($\pm$7.69)

F Additional Empirical Results

F.1 Additional Sequential Optimization Results

We include results for an additional synthetic benchmark: the **DTLZ2** problem from the MO literature [15] ($d = 6, M = 2$). Figure 6 shows that qEHVI outperforms all other baseline algorithms on the DTLZ2 in terms of sequential optimization performance with competitive wall times as shown in 3.
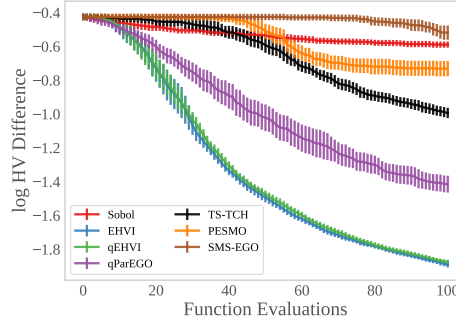
Figure 6: Optimization performance on the DTLZ2 synthetic function ($d = 6, M = 2$).

### F.2 Performance with Increasing Parallelism

Figure 7 shows that that the performance of qEHVI performance does not degrade substantially, whereas performance does degrade for qPAREGO and TS-TCH on some benchmark problems. We include results for all problems in Section 5 and Appendix F.1 as well as a **Constrained Branin-Currin** problem (which is described in Appendix E.2).
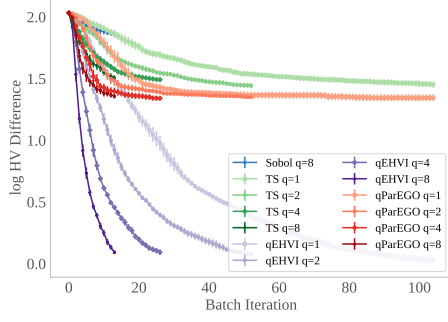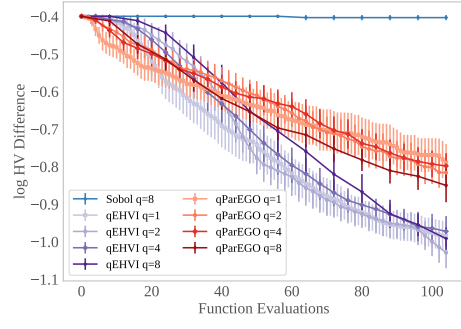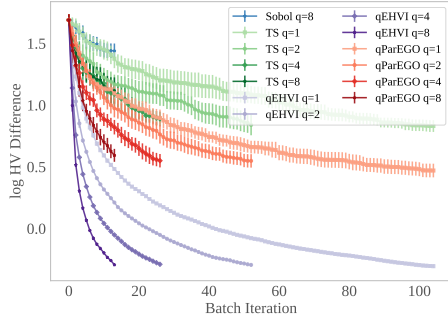
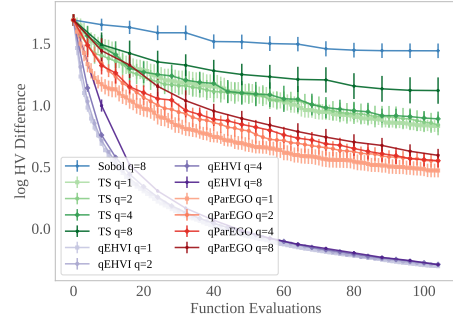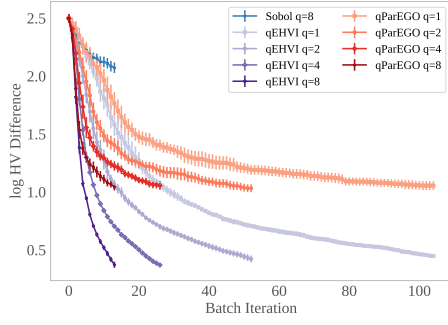(a) VEHICLESAFETY
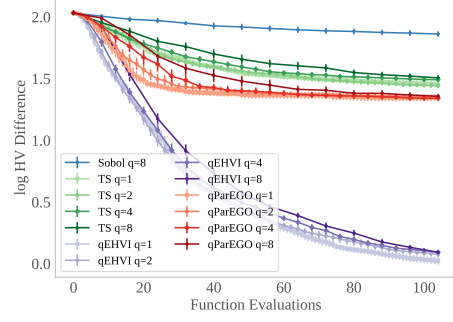
(b) VEHICLESAFETY
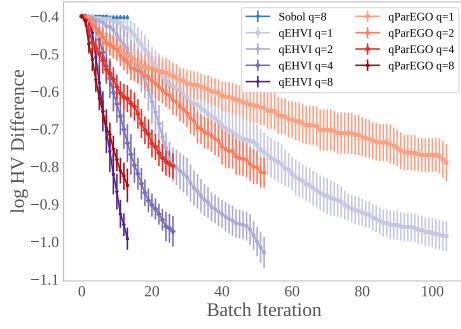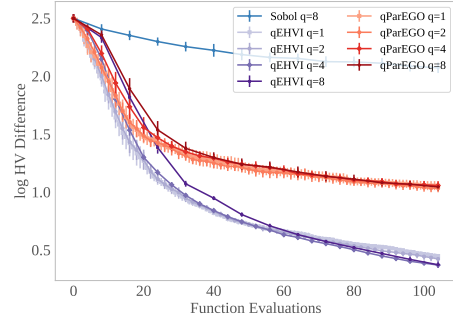
(c) C2DTLZ2

(d) C2DTLZ2

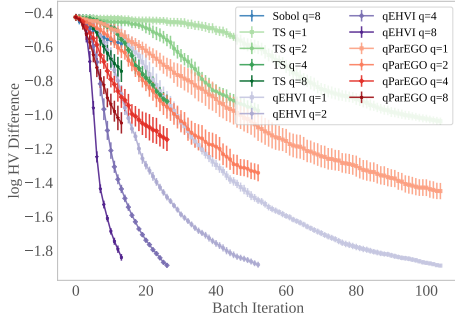(e) BRANINCURRIN

(f) BRANINCURRIN

Figure 7: Optimization performance of parallel acquisition functions over *batch BO iterations* (left) and *function evaluations* (right) for benchmark problems in Section 5.
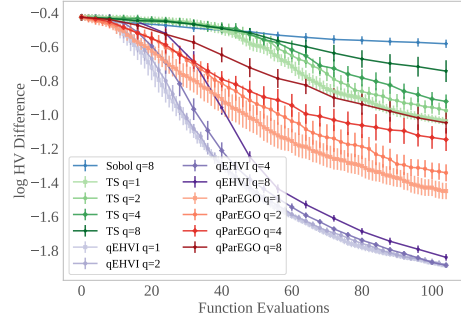
(a) CONSTRAINEDBRANINCURRIN

(b) CONSTRAINEDBRANINCURRIN

(c) DTLZ2 ($M = 2, d = 6$)

(d) DTLZ2 ($M = 2, d = 6$)

Figure 8: Optimization performance of parallel acquisition functions over *batch BO iterations* (left) and *function evaluations* (right) for additional benchmark problems.

F.3 Noisy Observations

Although neither qEHVI nor any variant of expected hypervolume improvement (to our knowledge) directly account for noisy observations, noisy observations are a practical challenge. We empirically evaluate the performance of all algorithms on a Branin-Currin function where observations have additive, zero-mean, iid Gaussian noise; the unknown standard deviation of the noise is set to be 1% of the range of each objective. Fig 9 shows that qEHVI performs favorably in the presence of noise, besting all algorithms including Noisy qPAREGO (qNParego) (described in Appendix E.1.2), PESMO and TS-TCH, all of which account for noise.
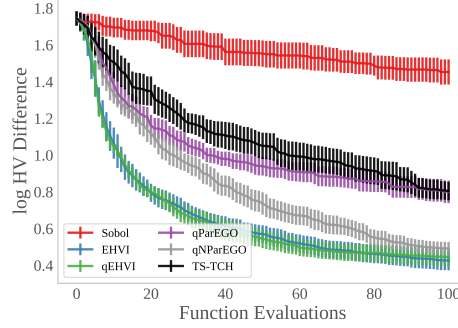

Figure 9: Sequential optimization performance on a noisy Branin-Currin problem.

F.4 Approximate Box Decompositions

EHVI becomes prohibitively computationally expensive in many scenarios with ≥ 4 objectives because of the wall time of partitioning the non-dominated space into disjoint rectangles [11]. Therefore, in addition to providing an exact binary partitioning algorithm, Couckuyt et al. [11] propose an approximation that terminates the partitioning algorithm when the new additional set of hyper-rectangles in the partitioning has a total hypervolume of less than a predetermined fraction ζ of the hypervolume dominated by the Pareto front. While qEHVI is guaranteed to be exact when an exact partitioning of the non-dominated space is used, qEHVI is agnostic to the partitioning algorithm used and is compatible with more scalable approximate methods.

We evaluate the performance of qEHVI with approximation of various fidelities ζ on DTLZ2 problems with 3 and 4 objectives (with $d = 6$). $\zeta = 0$ corresponds to an exact partitioning and the approximation is monotonically worse as ζ increases. Larger values of ζ degrade optimization performance (Figure 10), but can result in substantial speedups (Table 4). Even with coarser levels of approximation, qEHVI() performs better than qPAREGO with respect to log hypervolume difference, while achieving wall time improvements of 2-7x compared to exact qEHVI.
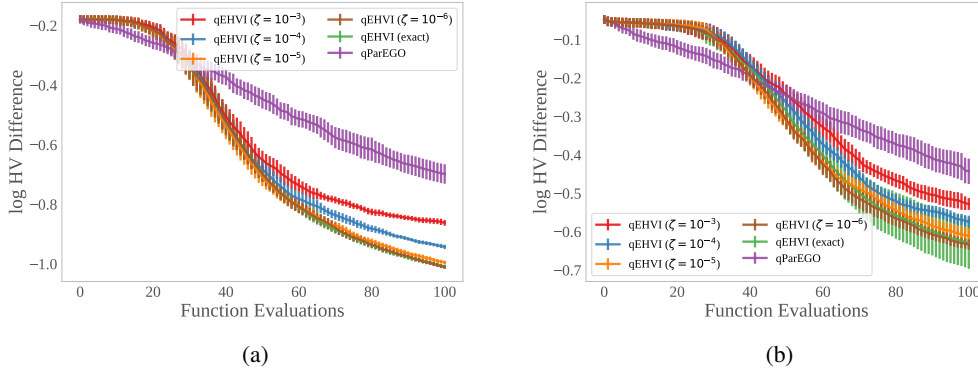

| (a) | (b) |

Figure 10: Optimization performance on DTLZ2 problems ($d = 6$) with approximate partitioning using various approximation levels ζ for (a) $M = 3$ objectives and (b) $M = 4$ objectives.

CPU	DTLZ2 ($M = 3$)	DTLZ2 ($M = 4$)
qPAREGO	5.86 ($\pm$0.51)	5.6 ($\pm$0.53)
qEHVI ($\zeta = 10^{-3}$)	6.89 ($\pm$0.41)	9.53 ($\pm$0.49)
qEHVI ($\zeta = 10^{-4}$)	9.83 ($\pm$0.9)	17.47 ($\pm$1.2)
qEHVI ($\zeta = 10^{-5}$)	18.99 ($\pm$2.72)	60.27 ($\pm$3.57)
qEHVI ($\zeta = 10^{-6}$)	37.9 ($\pm$7.47)	136.15 ($\pm$12.88)
qEHVI (EXACT)	45.52 ($\pm$9.83)	459.33 ($\pm$77.95)

Table 4: Acquisition function optimization wall time with approximate hypervolume computation, in seconds on a CPU (2x Intel Xeon E5-2680 v4 @ 2.40GHz). The mean and two standard errors are reported.

F.5 Acquisition Computation Time

Figure 11 show the acquisition computation time for different M and q. The inflection points corresponds to available processor cores becoming saturated. For large M an q on the GPU, memory becomes an issue, but we discuss ways of mitigating the issue in Appendix A.4.
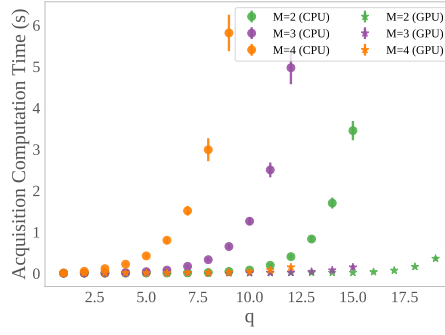

Figure 11: Acquisition computation time for different batch sizes q and numbers of objectives M (this excludes the time required to compute the acquisition function *given* box decomposition of the non-dominated space). This uses $N = 512$ MC samples, $d = 6$, $|\mathcal{P}| = 10$, and 20 training points. CPU time was measured on 2x Intel Xeon E5-2680 v4 @ 2.40GHz and GPU time was measured on a Tesla V100-SXM2-16GB GPU using 64-bit floating point precision. The mean and 2 standard errors over 1000 trials are reported.