Joint Entropy Search for Multi-Objective Bayesian Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Many real-world problems can be phrased as a multi-objective optimization problem, where the goal is to identify the best set of compromises between the competing objectives. Multi-objective Bayesian optimization (BO) is a sample efficient strategy that can be deployed to solve these vector-valued optimization problems where access is limited to a number of noisy objective function evaluations. In this paper, we propose a novel information-theoretic acquisition function for BO called Joint Entropy Search (JES), which considers the joint information gain for the optimal set of inputs and outputs. We present several analytical approximations to the JES acquisition function and also introduce an extension to the batch setting. We showcase the effectiveness of this new approach on a range of synthetic and real-world problems in terms of the hypervolume and its weighted variants.

1 Introduction

Bayesian optimization (BO) has demonstrated a lot of success in solving black-box optimization problems in various domains such as machine learning [64, 65, 78], chemistry [23, 31], robotics [7, 12] and clinical trials [53, 67]. The procedure works by maintaining a probabilistic model of the observed data in order to guide the optimization procedure into regions of interest. Specifically, at each iteration the black-box function is evaluated at one or more input locations that maximizes an acquisition function on the model. Implicitly, this function strikes a balance between exploring new areas and exploiting areas that have been shown to be promising. In this work, we consider the more general problem, where the black-box function of interest is vector-valued. This increases the difficulty of the problem because there are now many directions in which the objectives can be improved, in contrast to the single-objective setting where there is only one. Informally, the end goal of multi-objective optimization is to identify a collection of points that describe the best trade-off between the different objectives.

There are several ways to define an acquisition function for multi-objective BO. A popular strategy is random scalarization [43, 54], which works by transforming the multi-objective problem into a distribution of single-objective problems. These approaches are appealing because it enables the use of standard single-objective acquisition functions. A weakness of this approach is that it relies on random sampling to encourage exploration and therefore the performance of this method might suffer when the scale of the objectives is unknown a priori or when the input space is high-dimensional [16]. Another popular class of multi-objective acquisition functions are improvement-based. These strategies focus on improving a utility function over sets, for example the hypervolume indicator [14, 15, 22, 80] or the R2 indicator [20]. The main drawback of these approaches is that the performance of these methods are biased towards a single scalar utility criterion, which can be inadequate to assess the multi-objective aspects of the problem [83].

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

Instead of relying on scalarizations or an improvement-based criterion, this paper considers a probabilistic perspective where the goal of interest is to improve the posterior distribution at the optimal points. We propose a novel information-theoretic acquisition function called the Joint Entropy Search (JES), which assesses how informative an observation will be in learning more about the joint distribution of the optimal inputs and outputs. This acquisition function combines the advantages of existing information-theoretic acquisition functions, which focus solely on improving the posterior of either the optimal inputs [27, 29, 35] or the optimal outputs [4, 6, 68]. We connect JES with the existing information-theoretic acquisition functions by proving that it is an upper bound to these utilities.

Contributions and organization. In Section 2, we set-up the problem and introduce the novel JES acquisition function. In Section 3, we present a catalogue of conditional entropy estimates to approximate this utility and present a simple extension to the batch setting. All of these approximations are analytically available, which enables the use of gradient-based optimization. In Section 4, we present a discussion on the hypervolume indicator and explain how it can be a misleading performance criterion because it is sensitive to the scale of the objectives. We prove that information-theoretic approaches are naturally invariant to reparameterization of the objectives, which make them well-suited for multi-objective black-box optimization. For a more complete picture of performance, we propose a novel weighted hypervolume strategy (Appendix H), which allows us to assess the performance of a multi-objective algorithm over different parts of the objective space. In Section 5, we demonstrate the effectiveness of JES on some synthetic and real-life multi-objective problems. Finally in Section 6, we provide some concluding remarks. Additional results and proofs are presented in the Appendix.

2 Preliminaries

We consider the problem of maximizing a vector-valued function $f : \mathbb{X} \to \mathbb{R}^M$ over a bounded space of inputs $\mathbb{X} \subset \mathbb{R}^D$. To define the maximum $\max_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x})$, we appeal to the Pareto partial ordering in $\mathbb{R}^M$. For the rest of this paper, we will denote vectors by $\mathbf{y} = (y^{(1)}, \dots, y^{(M)}) \in \mathbb{R}^M$, the non-negative real numbers by $\mathbb{R}_{\geq 0}$ and diagonal matrices by $\mathrm{diag}(\cdot)$.

Pareto domination. We say a vector $\mathbf{y} \in \mathbb{R}^M$ weakly Pareto dominates another vector $\mathbf{y}' \in \mathbb{R}^M$ if it performs just as well in all objectives if not better: $\mathbf{y} \succeq \mathbf{y}' \iff \mathbf{y} - \mathbf{y}' \in \mathbb{R}^M_{\geq 0}$. Additionally, if the vectors are not equivalent, $\mathbf{y} \neq \mathbf{y}'$, then we say strict Pareto domination holds: $\mathbf{y} \succ \mathbf{y}' \iff \mathbf{y} - \mathbf{y}' \in \mathbb{R}^M_{\geq 0} \setminus \{\mathbf{0}_M\}$, where $\mathbf{0}_M$ is the M-dimensional zero vector. This binary relation can be further extended to define domination among sets. Let $A, B \subset \mathbb{R}^M$ be sets, if the set B lies in the weakly dominated region of A, namely $B \subset \mathbb{D}_{\preceq}(A) = \cup_{\mathbf{a}\in A}\{\mathbf{y} \in \mathbb{R}^M : \mathbf{y} \preceq \mathbf{a}\}$, then we say A weakly dominates B, denoted by $A \succeq B$. In addition, if it also holds that the dominated regions are not equal, $\mathbb{D}_{\preceq}(A) \neq \mathbb{D}_{\preceq}(B)$, we say strict Pareto domination holds, denoted by $A \succ B$.

Multi-objective optimization. The goal of multi-objective optimization is to identify the Pareto optimal set of inputs $\mathbb{X}^* = \arg\max_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x}) \subset \mathbb{X}$. The Pareto set is defined as the set of inputs whose objective vectors are not strictly Pareto dominated by another: $\mathbf{x}^* \in \mathbb{X}^* \iff \mathbf{x}^* \in \mathbb{X}$ and $\nexists \mathbf{x} \in \mathbb{X}$ such that $f(\mathbf{x}) \succ f(\mathbf{x}^*)$. The image of the Pareto set in the objective space $\mathbb{Y}^* = f(\mathbb{X}^*) = \max_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x})$ is called the Pareto front. For convenience of notation, we will denote the set of Pareto optimal input-output pairs by $(\mathbb{X}^*, \mathbb{Y}^*)$.

Bayesian Optimization is a sample efficient global optimization strategy, which relies on a probabilistic model in order to decide which points to query. In Appendix A.1, we present the pseudo-code for BO procedure—for more details see [11, 25, 63]. In this work, we will use independent Gaussian process priors [59] on each objective, $f^{(m)} \sim \mathrm{GP}(\mu_0^{(m)}, \Sigma_0^{(m)})$, where $\mu^{(m)} : \mathbb{X} \to \mathbb{R}$ is the mean function and $\Sigma^{(m)} : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is the covariance function for objective m. The observations at location $\mathbf{x} \in \mathbb{X}$ will be assumed to be corrupted with additive Gaussian noise, $\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{diag}(\boldsymbol{\sigma}(\mathbf{x}))$ denotes the observation noise with variance $\boldsymbol{\sigma}(\mathbf{x}) \in \mathbb{R}^M_{\geq 0}$. After n evaluations, we have a data set $D_n = \{(\mathbf{x}_t, \mathbf{y}_t)\}_{t=1,\dots,n}$. The posterior model $p(f|D_n)$ is a collection of independent Gaussian processes $f^{(m)}|D_n \sim \mathrm{GP}(\mu_n^{(m)}, \Sigma_n^{(m)})$. The explicit expressions for the mean and covariance are presented in Appendix A.2. The main focus of this work is on designing the acquisition function, $\alpha : \mathbb{X} \to \mathbb{R}$, which is used to select the inputs: $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathbb{X}} \alpha(\mathbf{x}|D_n)$.

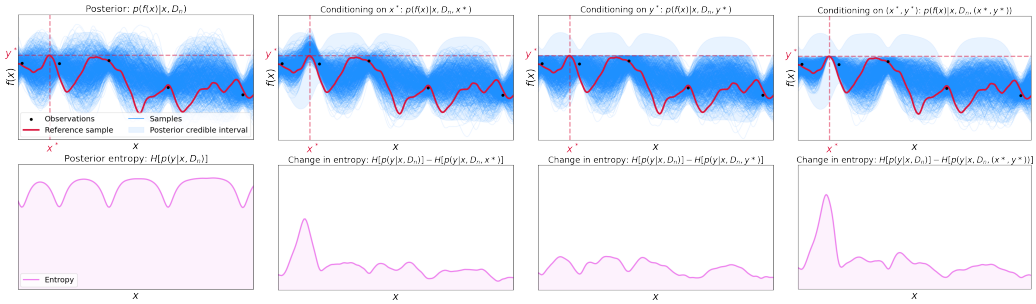


Figure 1: Comparison of the samples (top) and change in entropy (bottom) for the posterior and conditional distributions. The red line in the posterior plots denotes the reference sample that is used to obtain the maximizer x^* and maximum y^*, whilst the shaded blue region is the 95% credible interval of the posterior $p(f(x)|x, D_n)$. Conditioning on x^* reduces the entropy for all inputs according to how correlated it is with x^*. Conditioning on y^* reduces the entropy for all inputs according to the posterior probability that the objective surpasses y^*. Conditioning on (x^*, y^*) leads to a drop in entropy based on both the input correlation with x^* and the posterior probability of exceeding y^*.

Information-theoretic acquisition functions focus on maximizing the gain in information from the next observation and a property of the probabilistic model. Initial work in BO focussed on picking points to learn more about the distribution of the maximizer $p(\mathbb{X}^*|D_n)$. Specifically, the goal of interest was to maximize the mutual information between the observation $\mathbf{y}$ and the Pareto set $\mathbb{X}^*$ conditional on the current data set D_n:

$$\alpha^{\text{PES}}(\mathbf{x}|D_n) = \text{MI}(\mathbf{y}; \mathbb{X}^*|\mathbf{x}, D_n) = H[p(\mathbf{y}|\mathbf{x}, D_n)] - \mathbb{E}_{p(\mathbb{X}^*|D_n)}[H[p(\mathbf{y}|\mathbf{x}, D_n, \mathbb{X}^*)]] \tag{1}$$

where $H[p(\mathbf{x})] = -\int p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x}$ represents the differential entropy. This acquisition function is commonly referred to as predictive entropy search (PES) [35, 36, 62], but it was formerly[1] known as entropy search (ES) [34, 72]. Despite the importance of obtaining more information about the maximizer, the PES acquisition function is heavily dependent on the approximation of $p(\mathbf{y}|\mathbf{x}, D_n, \mathbb{X}^*)$, which is both computationally difficult to implement and optimize. This motivated researchers to consider a simpler scheme that focusses on learning more about the distribution of the maximum $p(\mathbb{Y}^*|D_n)$. The resulting acquisition function is known as the max-value entropy search (MES) [4, 40, 68, 74]:

$$\alpha^{\text{MES}}(\mathbf{x}|D_n) = \text{MI}(\mathbf{y}; \mathbb{Y}^*|\mathbf{x}, D_n) = H[p(\mathbf{y}|\mathbf{x}, D_n)] - \mathbb{E}_{p(\mathbb{Y}^*|D_n)}[H[p(\mathbf{y}|\mathbf{x}, D_n, \mathbb{Y}^*)]]. \tag{2}$$

Unlike PES, the conditional probability $p(\mathbf{y}|\mathbf{x}, D_n, \mathbb{Y}^*)$ arising in MES can be approximated and optimized more easily because some approximations lead to closed-form expressions. Despite the favourable properties of MES, the primary goal of interest is to identify the location of the maximizer $\mathbb{X}^*$ and not necessarily the value of the maximum $\mathbb{Y}^*$. To combine the advantages of both of these approaches, we propose the joint entropy search acquisition function, which focusses on learning more about the joint distribution of the optimal points $p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)$:

$$\begin{aligned} \alpha^{\text{JES}}(\mathbf{x}|D_n) &= \text{MI}(\mathbf{y}; (\mathbb{X}^*, \mathbb{Y}^*)|\mathbf{x}, D_n) \\ &= H[p(\mathbf{y}|\mathbf{x}, D_n)] - \mathbb{E}_{p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)}[H[p(\mathbf{y}|\mathbf{x}, D_n, (\mathbb{X}^*, \mathbb{Y}^*))]]. \end{aligned} \tag{3}$$

The JES acquisition function inherits the advantages of the PES and MES acquisition functions because it considers the knowledge learnt about the optimal points and is also simple to implement—more details in the next section. The following proposition shows that we can also interpret JES as an upper bound to both the PES and MES acquisition function.

Proposition 1 *The JES is an upper bound to any convex combination of the PES and MES acquisition functions:* $\alpha^{\text{JES}}(\mathbf{x}|D_n) \geq \beta\alpha^{\text{PES}}(\mathbf{x}|D_n) + (1-\beta)\alpha^{\text{MES}}(\mathbf{x}|D_n)$, for any $\beta \in [0, 1]$.

[1] The difference in the naming convention stems solely from the approximation strategy used to estimate the mutual information. At a high level, ES applies expectation propagation [49] to estimate $p(\mathbb{X}^*|D_n \cup \{\mathbf{x}, \mathbf{y}\})$, whilst PES applies expectation propagation to estimate $p(\mathbf{y}|\mathbf{x}, D_n, \mathbb{X}^*)$.

3 Approximating JES

In this section, we present several approximations to the JES acquisition function (3) and a simple extension to the batch setting. The first term in the JES criterion (3) is the entropy of a multivariate normal distribution:

$$H[p(\mathbf{y}|\mathbf{x}, D_n)] = \frac{M}{2}\log(2\pi e) + \frac{1}{2}\sum_{m=1}^{M}\log(\Sigma_n^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})). \tag{4}$$

The second term is an intractable expectation which is approximated by drawing Monte Carlo samples from $p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)$. The conditional entropy $H[p(\mathbf{y}|\mathbf{x}, D_n, (\mathbb{X}^*, \mathbb{Y}^*))]$ is also an intractable quantity which has to be estimated. The overall approximation of (3) will take the form

$$\hat{\alpha}^{\text{JES}}(\mathbf{x}|D_n) = H[p(\mathbf{y}|\mathbf{x}, D_n)] - \frac{1}{S}\sum_{s=1}^{S} h((\mathbb{X}_s^*, \mathbb{Y}_s^*); \mathbf{x}, D_n), \tag{5}$$

where h denotes the conditional entropy estimate and $(\mathbb{X}_s^*, \mathbb{Y}_s^*) \sim p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)$ are the Monte Carlo samples. The distribution $p(\mathbf{y}|\mathbf{x}, D_n, (\mathbb{X}^*, \mathbb{Y}^*))$ is very challenging to work with because it enforces the global optimality condition that the function lies below the Pareto front $f(\mathbb{X}) \preceq \mathbb{Y}^*$. Instead of enforcing global optimality, we make the common simplifying assumption as in [50, 68, 74] and only enforce the optimality condition at the considered location: $f(\mathbf{x}) \preceq \mathbb{Y}^*$. By applying Bayes' theorem, the resulting density of interest becomes

$$p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*) = \frac{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} p(\mathbf{y}|\mathbf{x}, D_{n*}), \tag{6}$$

where we have denoted the augmented data sets by $D_{n*} = D_n \cup (\mathbb{X}^*, \mathbb{Y}^*)$ and $D_{n+} = D_{n*} \cup \{(\mathbf{x}, \mathbf{y})\}$. We will refer to the quantity $p(f(\mathbf{x}) \preceq \mathbb{Y}^*)$ as the cumulative distribution function (CDF). The following lemma shows that this CDF can be computed analytically when the set $\mathbb{Y}^* \subset \mathbb{R}^M$ is discrete. This result can be derived by first partitioning the region of integration, $\mathbb{D}_{\preceq}(\mathbb{Y}^*) = \cup_{\mathbf{y}^* \in \mathbb{Y}^*}\{\mathbf{z} \in \mathbb{R}^M : \mathbf{z} \preceq \mathbf{y}^*\}$, into a collection of hyperrectangle subsets and then summing up the individual contributions—see Figure 2 for a visual. This partition can be computed using an incremental approach of (Algorithm 1 of [46]), which has a cost of $O(|\mathbb{Y}^*|^{\lfloor M/2 \rfloor + 1})$.

Lemma 1 *Let $\mathbb{Y}^* \subset \mathbb{R}^M$ be a finite set and $\mathbf{z} \sim N(\mathbf{a}, diag(\mathbf{b}))$ be an M-dimensional multivariate normal with mean $\mathbf{a} \in \mathbb{R}^M$ and variances $\mathbf{b} \in \mathbb{R}^M_{\geq 0}$. Let $\mathbb{D}_{\preceq}(\mathbb{Y}^*) = \bigcup_{j=1}^{J} B_j = \bigcup_{j=1}^{J}\prod_{m=1}^{M}(l_j^{(m)}, u_j^{(m)}]$ be the box decomposition of the dominated space, then*

$$p(\mathbf{z} \preceq \mathbb{Y}^*) = \sum_{j=1}^{J}\prod_{m=1}^{M}\left[\Phi\left(\frac{u_j^{(m)} - a^{(m)}}{\sqrt{b^{(m)}}}\right) - \Phi\left(\frac{l_j^{(m)} - a^{(m)}}{\sqrt{b^{(m)}}}\right)\right]. \tag{7}$$

Algorithm 1: Joint Entropy Search (JES).

Input : A candidate $\mathbf{x}$; the data set D_n.

`// Cached variables are coloured in blue.`

Compute the initial entropy $h_0 = H[p(\mathbf{y}|\mathbf{x}, D_n)]$.
for $s = 1, \ldots, S$ **do**
  Sample a path $f_s \sim p(f|D_n)$.
  Compute the Pareto optimal points $\mathbb{X}_s^* = \arg\max_{\mathbf{x}' \in \mathbb{X}} f_s(\mathbf{x}')$ and $\mathbb{Y}_s^* = f_s(\mathbb{X}_s^*)$.
  Compute the box decomposition $\mathbb{D}_{\preceq}(\mathbb{Y}_s^*) = \bigcup_{j=1}^{J} B_j$.
  Compute the conditional $p(f|D_n \cup (\mathbb{X}_s^*, \mathbb{Y}_s^*))$.
  Compute the estimate $h_s = h((\mathbb{X}_s^*, \mathbb{Y}_s^*); \mathbf{x}, D_n)$.
end
return $\hat{\alpha}^{\text{JES}}(\mathbf{x}|D_n) = h_0 - \frac{1}{S}\sum_{s=1}^{S} h_s$.

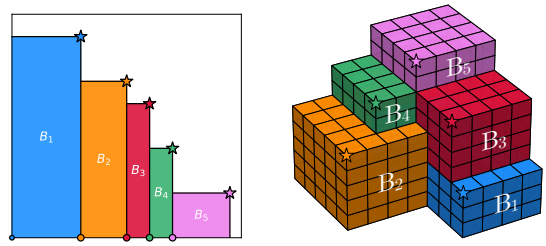


Figure 2: Box decompositions for a two-dimensional and three-dimensional Pareto front.

3.1 Estimating the conditional entropy

The entropy of (6) can be written as an M-dimensional expectation over the multivariate normal distribution $p(\mathbf{y}|\mathbf{x}, D_{n*}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{n*}(\mathbf{x}), \boldsymbol{\Sigma}_{n*}(\mathbf{x}, \mathbf{x}))$:

$$\begin{aligned} &H[p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)] \\ &= -\mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}\left[\frac{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} \log\left(\frac{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} p(\mathbf{y}|\mathbf{x}, D_{n*})\right)\right]. \end{aligned} \tag{8}$$

To simplify the notation, we define the m-th standardized value by $\gamma_m(z) = (z - \mu_{n*}^{(m)}(\mathbf{x}))/\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x})}$ for any scalar $z \in \mathbb{R}$. Using this function together with Lemma 1, we denote the cumulative distribution $p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})$ by $W = \sum_{j=1}^{J} W_j = \sum_{j=1}^{J} \prod_{m=1}^{M} W_{j,m}$, where $W_{j,m} = \Phi(\gamma_m(u_j^{(m)})) - \Phi(\gamma_m(l_j^{(m)}))$ are the differences appearing in (7). Moreover, we denote the differences of the first derivative of $W_{j,m}$ (with respect to γ_m) by $G_{j,m} = \phi(\gamma_m(u_j^{(m)})) - \phi(\gamma_m(l_j^{(m)}))$ and the negative of the second derivative by $V_{j,m} = \gamma_m(u_j^{(m)})\phi(\gamma_m(u_j^{(m)})) - \gamma_m(l_j^{(m)})\phi(\gamma_m(l_j^{(m)}))$, where ϕ is the probability density function of a standard normal distribution. In the setting where the observation noise is zero, the conditional distribution is a truncated multivariate normal, which is known to have the following analytical equation for the entropy:

Proposition 2 *(Theorem 3.1. in [68]) Under the modelling set-up outlined in Section 2, if $\mathbf{x} \in \mathbb{X}$ is an input with zero observation variance, $\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{0}_M$, then the entropy of the truncated multivariate normal distribution $p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)$ is given by*

$$\begin{aligned} &H[p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)] \\ &= \frac{M}{2}\log(2\pi e) + \frac{1}{2}\sum_{m=1}^{M}\log(\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x})) + \log W - \frac{1}{2W}\sum_{j=1}^{J} W_j \sum_{m=1}^{M}\frac{V_{j,m}}{W_{j,m}}. \end{aligned}$$

Consequently, we define the JES-0 entropy estimate by the equation

$$\begin{aligned} &h^{\text{JES-0}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n) \\ &= \frac{M}{2}\log(2\pi e) + \frac{1}{2}\sum_{m=1}^{M}\log(\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x}) + \sigma^{(m)}(\mathbf{x})) + \log W - \frac{1}{2W}\sum_{j=1}^{J} W_j \sum_{m=1}^{M}\frac{V_{j,m}}{W_{j,m}}. \end{aligned} \tag{9}$$

Notice that we have included an additional observation noise term $\sigma^{(m)}(\mathbf{x})$ into the logarithm of the variance. This term will cancel out with the term in the initial entropy (4) in both the noiseless and noisy setting. Intuitively, this is the implicit noise correction when applying Proposition 2 in settings where the observation noise is non-zero. We will see in Section 5 that this correction is quite effective in practice.

In the noisy setting, the distribution of interest is a type of multivariate skew normal distribution, which is known to not have an analytical form for the entropy [1]. As a result, we propose two approximation strategies to estimate this entropy. The first strategy is to approximate the integral using Monte Carlo. The details of the Monte Carlo estimate $h^{\text{JES-MC}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n)$ is

described in Appendix D. The second strategy is to directly approximate the distribution with one that exhibits an analytical entropy. We consider the most obvious choice, which is a multivariate normal distribution with the same first two moments. The same strategy was proposed in [50] for the single-objective multi-fidelity MES acquisition function. By a standard result (Chapter 12 of [13]), the entropy of this approximating distribution is actually an upper bound for the entropy of interest: $H[p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)] \leq \frac{M}{2}\log(2\pi e) + \frac{1}{2}\log\det \mathbb{V}\text{ar}(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)$. The following result shows that the these central moments can be computed analytically.

Proposition 3 *Under the modelling set-up outlined in Section 2, for an input $\mathbf{x} \in \mathbb{X}$ the first and second central moment of $p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)$ are*

$$\mathbb{E}[y^{(m)}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*] = \mu_{n*}^{(m)}(\mathbf{x}) - \frac{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}}{W}\sum_{j=1}^{J} W_j \frac{G_{j,m}}{W_{j,m}}$$

and

$$\text{Cov}\left(y^{(m)}, y^{(m')}\middle|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right)$$
$$= \begin{cases} \frac{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}\sqrt{\Sigma_{n*}^{(m')}(\mathbf{x},\mathbf{x})}}{W}\sum_{j=1}^{J} W_j \frac{G_{j,m}}{W_{j,m}}\left(\frac{G_{j,m'}}{W_{j,m'}} - \frac{1}{W}\sum_{j'=1}^{J} W_{j'}\frac{G_{j',m'}}{W_{j',m'}}\right), & m \neq m'; \\ \Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x}) - \frac{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}{W}\left(\sum_{j=1}^{J} W_j \frac{V_{j,m}}{W_{j,m}} + \frac{1}{W}\left(\sum_{j=1}^{J} W_j \frac{G_{j,m}}{W_{j,m}}\right)^2\right), & m = m'. \end{cases}$$

As an upper bound on the conditional entropy leads to a lower bound on the mutual information. We define the JES-LB entropy estimate as follows:

$$h^{\text{JES-LB}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n) = \frac{M}{2}\log(2\pi e) + \frac{1}{2}\log\det \mathbb{V}\text{ar}(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*). \tag{10}$$

We could obtain a further lower bound by ignoring the off-diagonal terms in the covariance matrix. We dub the resulting approximation as the JES-LB2 entropy estimate:

$$h^{\text{JES-LB2}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n) = \frac{M}{2}\log(2\pi e) + \frac{1}{2}\sum_{m=1}^{M}\log \mathbb{V}\text{ar}(y^{(m)}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*). \tag{11}$$

Empirically, we observe that the lower bound JES approximations are not necessarily tight for all inputs $\mathbf{x} \in \mathbb{X}$. Nevertheless, the maximizer of the approximate acquisition function is often close to the maximizer of the exact JES—see Appendix G for a visualization of a single-objective example.

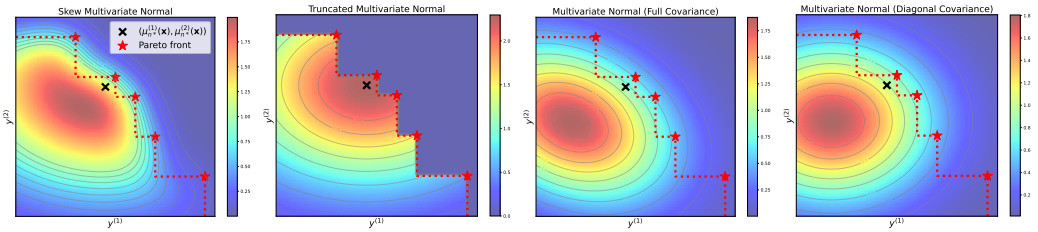
Remark. All of the conditional entropy estimates introduced here can be applied to the MES acquisition function. The only difference is that we no longer apply the conditioning step (line 6 of Algorithm 1) because we are interested in estimating $H[p(\mathbf{y}|\mathbf{x}, D_n, f(\mathbf{x}) \preceq \mathbb{Y}^*)]$ as opposed to (8). Consequently, the MES acquisition function is cheaper to evaluate because the cost of a evaluating the posterior variance at a single input is $O(n^2)$, whereas JES incurs a cost of $O((n + |\mathbb{Y}^*|)^2)$—more details are presented in the cost analysis in Appendix F.

3.2 Batch evaluations

Evaluating the JES acquisition functions for a batch of points $\mathbf{x}^{[1:q]} = (\mathbf{x}^{[1]}, \ldots, \mathbf{x}^{[q]}) \in \mathbb{X}^q$ is expensive because the entropy estimates now depends on the q-dimensional normal CDF and its derivatives. To circumvent this issue, we follow the example of [50] and propose a suboptimal batch approach by upper bounding the expensive joint conditional entropy term by the sum of the individual entropies: $H[p(\mathbf{y}^{[1:q]}|\mathbf{x}^{[1:q]}, D_{n*}, f(\mathbb{X}) \preceq \mathbb{Y}^*)] \leq \sum_{i=1}^{q} H[p(\mathbf{y}^{[i]}|\mathbf{x}^{[i]}, D_{n*}, f(\mathbb{X}) \preceq \mathbb{Y}^*)]$. The resulting q-batch lower bound JES estimate is given by

$$\hat{\alpha}^{q\text{LB-JES}}(\mathbf{x}^{[1:q]}|D_n) = H[p(\mathbf{y}^{[1:q]}|\mathbf{x}, D_n)] - \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{q} h((\mathbb{X}_s^*, \mathbb{Y}_s^*); \mathbf{x}^{[i]}, D_n), \tag{12}$$

where $h \in \{h^{\text{JES-0}}, h^{\text{JES-LB}}, h^{\text{JES-LB2}}, h^{\text{JES-MC}}\}$ are the conditional entropy estimates and $H[p(\mathbf{y}^{[1:q]}|\mathbf{x}, D_n)] = \frac{M}{2}\log(2\pi e) + \frac{1}{2}\sum_{m=1}^{M}\log\det(\Sigma_n^{(m)}(\mathbf{x}^{[1:q]}, \mathbf{x}^{[1:q]}) + \text{diag}(\sigma^{(m)}(\mathbf{x}^{[1:q]})))$ is



(a) (b) (c) (d)

Figure 3: Comparison of the density approximations to the skew multivariate normal distribution $p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)$ shown in (a) for a single input $\mathbf{x} \in \mathbb{X}$ and sample Pareto front $\mathbb{Y}^*$. A zero noise assumption leads to the truncated multivariate normal approximation shown (b), whilst a moment matching approach leads to the multivariate normal approximations in (c) and (d).

the initial entropy. This acquisition function is defined over a qD-dimensional space, which becomes more difficult to optimize as q increases. Alternatively, we can maximize this function greedily by sequentially selecting the best input conditioned on the previously chosen points. This greedy procedure satisfies an e^{-1} regret bound when the acquisition function is submodular [75]. In Appendix E, we show that this batch acquisition function is indeed submodular.

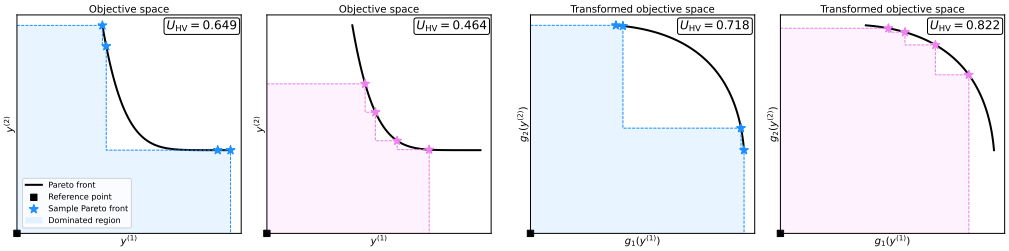
4 Performance criteria

In multi-objective optimization, the most common way to measure performance is by comparing the approximate Pareto set $\hat{\mathbb{X}}^*$ against the optimal Pareto set $\mathbb{X}^*$ in the objective space: $d(f(\hat{\mathbb{X}}^*), f(\mathbb{X}^*))$ where $d : 2^{\mathbb{R}^M} \times 2^{\mathbb{R}^M} \to \mathbb{R}$ is a function that measures the discrepancy between the sets of objective vectors. Existing work in multi-objective BO mainly focusses on the hypervolume (HV) discrepancy, $d_{\text{HV}}(A, B) = |U_{\text{HV}}(A) - U_{\text{HV}}(B)|$, where the HV indicator, $U_{\text{HV}}(A) = \int_{\mathbb{R}^M} \mathbb{I}[\mathbf{r} \preceq \mathbf{z} \preceq A] d\mathbf{z}$, is defined as the volume between a reference point[2] $\mathbf{r} \in \mathbb{R}^M$ and a set $A \subset \mathbb{R}^M$. An attractive feature of the HV indicator is that it is Pareto complete (or compliant) in the sense that a better set will lead to a larger HV [83]: $A \succ B \implies U_{\text{HV}}(A) > U_{\text{HV}}(B)$. The reverse implication known as Pareto compatability does not hold for the HV indicator [83]. In other words, the HV can be used to discriminate between sets where one dominates another, but it cannot be relied upon when the sets are incomparable. Not all incomparable sets are treated equally by the HV indicator [84]. For instance Figure 4a shows an example where the HV indicator places more emphasis on the end points of the Pareto front. On other hand, if we apply a monotonically increasing transformation $g_m : \mathbb{R} \to \mathbb{R}$ to each objective, the Pareto set will not change, whereas the HV comparison will (Figure 4b). Implicitly, the HV indicator assumes that a linear change in one objective is equivalent to a linear change in another. This assumption might not necessarily reflect the decision maker's outlook and this is something that is typically overlooked when designing and benchmarking multi-objective optimization algorithms. The following result shows that information-theoretic acquisitions function are in fact agnostic to the choice of parametrization.

Proposition 4 *The information-theoretic acquisition functions α^{PES}, α^{MES} and α^{JES} are invariant to reparametrizations of the objective space that are consistent with the Pareto ordering relations. For example, $\alpha^{\text{JES}}(\mathbf{x}|D_n) = \text{MI}(\mathbf{y}; (\mathbb{X}^*, \mathbb{Y}^*)|D_n) = \text{MI}(g(\mathbf{y}); (\mathbb{X}^*, g(\mathbb{Y}^*))|D_n)$, where the $g_m : \mathbb{R} \to \mathbb{R}$ is a strictly monotonically increasing function acting only on the m-th objective.*

To benchmark the algorithms, we use both the standard HV discrepancy (Section 5) and the HV discrepancy under different parameterizations (Appendix I). To easily obtain a family of parameterizations, we devise a novel weighting approach in Appendix H, which exploits the fact that the HV indicator can be written as an expectation over a uniform distribution on the $(M-1)$-dimensional hypercube [19, 81]. We observe that it is possible to assess the performance at different locations of the objective space by using alternate distributions over hypercube. In our experiments, we found

[2]The reference point is typically chosen to be the nadir, which is the vector comprised of the worst values for each objective $r^{(m)} = \min_{\mathbf{x} \in \mathbb{X}} f^{(m)}(\mathbf{x})$ for $m = 1, \ldots, M$.



(a) (b)

Figure 4: Comparison of the HV of two sample Pareto fronts in (a) the standard objective space $\mathbf{y} = (y^{(1)}, y^{(2)})$ and (b) the transformed objective space $g(\mathbf{y}) = (g_1(y^{(1)}), g_2(y^{(2)}))$ described in Appendix H. The HV indicator prefers a different set depending on the choice of parametrization.

that the performance of each algorithm changed with regards to the choice of parametrization, but the JES approaches tended to be the strongest performers throughout.

5 Experiments

We empirically evaluate the JES acquisition function on a range of synthetic and real-world benchmark problems. We compare this approach with some popular acquisition functions in multi-objective BO: TSEMO [10], ParEGO [43], NParEGO [15], EHVI [14], NEHVI [15], PES [27, 29] and MES-0 [68]. We have also included the MES-LB, MES-LB2 and MES-MC acquisition functions, which can be easily derived from the conditional entropy estimates that we developed here. All algorithms are based on the open source Python library BoTorch [3], which uses features from GPyTorch [26] for Gaussian process regression and PyTorch [56] for automatic differentiation. All experiments are repeated using 100 different initial seeds and we generate the Pareto set recommendation $\hat{\mathbb{X}}^*$ of 50 points by maximizing the posterior mean using a multi-objective solver (NSGA2 [18] from the Pymoo library [8]). The complete details of the experiments are outlined in Appendix I, whilst the code is available at `https://github.com/.../JES`.

5.1 Benchmarks

Synthetic benchmark. We consider the ZDT2 [18] benchmark with $D = 6$ inputs and $M = 2$ objectives. We corrupt the observations with additive Gaussian noise with zero-mean and standard deviation set to approximately 10% of the objective ranges.

Chemical reaction. This benchmark considers a nucleophilic aromatic substitution reaction (SnAr) between 2,4-difluoronitrobenzene and pyrrolidine in ethanol to produce a mixture of a desired product and two side-products [41]. The design space comprises of $D = 4$ components relating to the initial conditions. The goal is to optimize $M = 2$ objectives, namely the space time yield and the environmental impact. We apply a logarithm transform to the objectives and contaminate the observations with additive Gaussian noise with zero-mean and standard deviation set to approximately 3% of the resulting objective ranges in order to emulate a potential real-world scenario.

Pharmaceutical manufacturing. This problem is concerned with optimizing the Penicillin production process outlined in [47]. The design space is made up of $D = 7$ elements that control the initial condition of the reactions. The goal is to optimize $M = 3$ objectives, which relates to the yield, the amount of carbon dioxide released and the time to ferment. We include additive zero-mean Gaussian noise with a standard deviation set to approximately 1% of the objective ranges.

Marine design. This problem considers optimizing a family of bulk carriers subject to the constraints imposed for ships travelling through the Panama Canal [55, 61]. The design space is made up of $D = 6$ variables that determine the architecture of the carriers. The goal of this problem is to maximize the annual cargo, whilst minimizing the transportation cost and the ship weight subject to some design constraints. We consider the reformulation in [71], which converts the constraints

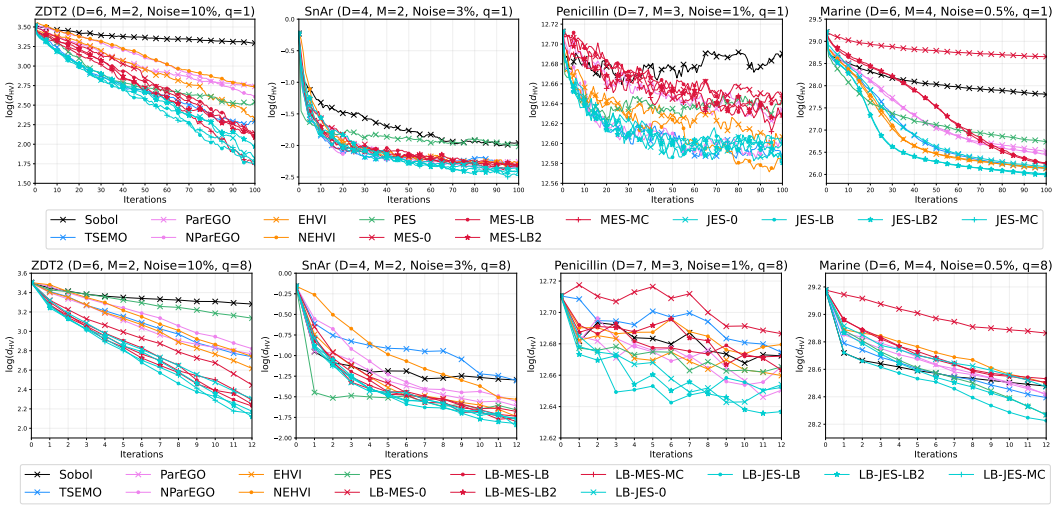


Figure 5: A comparison of the mean logarithm HV discrepancy over 100 runs for the four benchmark problems with a batch size of $q = 1$ and $q = 8$.

into another objective. For this reformulated $M = 4$ objective problem, we corrupt the observations with additive zero-mean Gaussian noise with standard deviation set to approximately 0.5% of the objective ranges.

5.2 Results and discussion

We present the mean log HV discrepancy for both the sequential and batch experiments in Figure 5. Appendix I contains additional figures describing the variation of the HV (and weighted HV) among the experiments, the wall times and some visual snapshots of the Pareto front approximations. Overall we see that the JES acquisition function is superior to existing methods in terms of the HV (and weighted HV) criterion on these benchmark problems. This gain in performance does come at the cost of additional wall time, which could potentially be alleviated by performing some routines in parallel—further details in Appendix I.6. Among the different conditional entropy approximations, we observe that the novel lower bound approximations offers the best trade-off between the HV performance and the wall time for both the JES and MES.

The improvement and scalarization based acquisition functions are faster to optimize than JES but they performed marginally worse in these sets of experiments since they tend to be more greedy. Specifically, these algorithms focus of querying high-performing points as opposed to the points that are the most informative for the posterior. As discussed in [15], this feature might be desirable for certain domains where the end goal is to make a decision from the final data set D_N as opposed to using the final model $p(f|D_N)$. Nevertheless, improvement-based acquisition functions inherently place a preference on certain parts of the Pareto front and therefore some additional thought has to be taken before applying these methods. Proposition 4 and the empirical results motivates the usage of information-theoretic acquisition functions in the truly black-box setting where all points on the Pareto front are equally desirable a priori.

6 Conclusion

We introduced JES, a novel information-theoretic acquisition function for multi-objective BO. To approximate this acquisition function, we presented several approximations to the conditional entropy and also a simple extension for the batch setting. Experimental results suggest that JES is very competitive with existing acquisition functions in terms of the HV discrepancy and its weighted variants. The main limitation of the JES acquisition function is that it relies on routines such as box decompositions and multi-objective optimization, which can be expensive to execute for very large problems. Future work could focus on improving the scalability of these information-theoretic methods and extending it to more general settings—see Appendix J.

References

[1] Reinaldo B. Arellano-Valle, Javier E. Contreras-Reyes, and Marc G. Genton. Shannon Entropy and Mutual Information for Multivariate Skew-Elliptical Distributions. *Scandinavian Journal of Statistics*, 40(1): 42–62, 2013. DOI: 10.1111/j.1467-9469.2011.00774.x.

[2] Adelchi Azzalini. A Class of Distributions Which Includes the Normal Ones. *Scandinavian Journal of Statistics*, 12(2):171–178, 1985.

[3] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 21524–21538, 2020.

[4] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value Entropy Search for Multi-Objective Bayesian Optimization. In *Advances in Neural Information Processing Systems*, pages 7825–7835, 2019.

[5] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Multi-Fidelity Multi-Objective Bayesian Optimization: An Output Space Entropy Search Approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10035–10043, April 2020. DOI: 10.1609/aaai.v34i06.6560.

[6] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Output Space Entropy Search Framework for Multi-Objective Bayesian Optimization. *Journal of Artificial Intelligence Research*, 72:667–715, November 2021. DOI: 10.1613/jair.1.12966.

[7] Felix Berkenkamp, Andreas Krause, and Angela P. Schoellig. Bayesian optimization with safety Constraints: Safe and automatic parameter tuning in robotics. *Machine Learning*, June 2021. DOI: 10.1007/s10994-021-06019-1.

[8] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, 8: 89497–89509, 2020. DOI: 10.1109/ACCESS.2020.2990567.

[9] Salomon Bochner, Monotonic Functions, Stieltjes Integrals, Harmonic Analysis, Morris Tenenbaum, and Harry Pollard. *Lectures on Fourier Integrals.* 1959.

[10] Eric Bradford, Artur M. Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *Journal of Global Optimization*, 71(2): 407–438, June 2018. DOI: 10.1007/s10898-018-0609-2.

[11] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *arXiv:1012.2599*, December 2010.

[12] Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1):5–23, February 2016. DOI: 10.1007/s10472-015-9463-9.

[13] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Hoboken, N.J, 2nd edition edition, September 2006.

[14] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 9851–9864, 2020.

[15] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian Optimization of Multiple Noisy Objectives with Expected Hypervolume Improvement. In *Advances in Neural Information Processing Systems*, page 14, 2021.

[16] Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-Objective Bayesian Optimization over High-Dimensional Search Spaces. *arXiv:2109.10964*, September 2021.

[17] Samuel Daulton, Sait Cakmak, Maximilian Balandat, Michael A. Osborne, Enlu Zhou, and Eytan Bakshy. Robust Multi-Objective Bayesian Optimization Under Input Noise. *arXiv:2202.07549*, February 2022.

[18] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002. DOI: 10.1109/4235.996017.

[19] Jingda Deng and Qingfu Zhang. Approximating Hypervolume and Hypervolume Contributions Using Polar Coordinate. *IEEE Transactions on Evolutionary Computation*, 23(5):913–918, October 2019. DOI: 10.1109/TEVC.2019.2895108.

[20] André Deutz, Michael Emmerich, and Kaifeng Yang. The Expected R2-Indicator Improvement for Multi-objective Bayesian Optimization. In *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 359–370, Cham, 2019. DOI: 10.1007/978-3-030-12598-1_29.

[21] Kun Dong, David Eriksson, Hannes Nickisch, David Bindel, and Andrew G Wilson. Scalable Log Determinants for Gaussian Process Kernel Learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[22] Michael T. M. Emmerich, Kyriakos C. Giannakoglou, and Boris Naujoks. Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, August 2006. DOI: 10.1109/TEVC.2005.859463.

[23] Kobi C. Felton, Jan G. Rittig, and Alexei A. Lapkin. Summit: Benchmarking Machine Learning Methods for Reaction Optimisation. *Chemistry–Methods*, 1(2):116–122, 2021. DOI: 10.1002/cmtd.202000051.

[24] Daniel Fernández-Sánchez, Eduardo C. Garrido-Merchán, and Daniel Hernández-Lobato. Improved Max-value Entropy Search for Multi-objective Bayesian Optimization with Constraints. *arXiv:2011.01150*, April 2021.

[25] Peter I. Frazier. A Tutorial on Bayesian Optimization. *arXiv:1807.02811*, July 2018.

[26] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[27] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints. *Neurocomputing*, 361:50–68, October 2019. DOI: 10.1016/j.neucom.2019.06.025.

[28] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing*, 380:20–35, March 2020. DOI: 10.1016/j.neucom.2019.11.004.

[29] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Parallel Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints. *arXiv:2004.00601*, July 2021.

[30] Michael A. Gelbart, Jasper Snoek, and Ryan P. Adams. Bayesian Optimization with Unknown Constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pages 250–259, Arlington, Virginia, USA, July 2014.

[31] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2):268–276, February 2018. DOI: 10.1021/acscentsci.7b00572.

[32] Robert B. Gramacy, Annie Sauer, and Nathan Wycoff. Triangulation candidates for Bayesian optimization. *arXiv:2112.07457*, December 2021.

[33] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825): 357–362, September 2020. DOI: 10.1038/s41586-020-2649-2.

[34] Philipp Hennig and Christian J. Schuler. Entropy Search for Information-Efficient Global Optimization. *Journal of Machine Learning Research*, 13(57):1809–1837, 2012.

[35] Daniel Hernandez-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive Entropy Search for Multi-objective Bayesian Optimization. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1492–1501, June 2016.

[36] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive Entropy Search for Efficient Global Optimization of Black-box Functions. In *Advances in Neural Information Processing Systems*, volume 27, pages 918–926, 2014.

[37] Jose Miguel Hernandez-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive Entropy Search for Bayesian Optimization with Unknown Constraints. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1699–1707, June 2015.

[38] José Miguel Hernández-Lobato, Michael A. Gelbart, Ryan P. Adams, Matthew W. Hoffman, and Zoubin Ghahramani. A General Framework for Constrained Bayesian Optimization using Information-based Search. *Journal of Machine Learning Research*, 17(160):1–53, 2016.

[39] Fred J. Hickernell, Christiane Lemieux, and Art B. Owen. Control Variates for Quasi-Monte Carlo. *Statistical Science*, 20(1):1–31, February 2005. DOI: 10.1214/088342304000000468.

[40] Matthew W Hoffman and Zoubin Ghahramani. Output-Space Predictive Entropy Search for Flexible Global Optimization. *NIPS workshop on Bayesian Optimization*, page 5, 2015.

[41] Christopher A. Hone, Nicholas Holmes, Geoffrey R. Akien, Richard A. Bourne, and Frans L. Muller. Rapid multistep kinetic model generation from transient flow data. *Reaction Chemistry & Engineering*, 2 (2):103–108, April 2017. DOI: 10.1039/C6RE00109B.

[42] Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity Bayesian Optimisation with Continuous Approximations. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1799–1808, July 2017.

[43] 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, February 2006. DOI: 10.1109/TEVC.2005.851274.

[44] Andreas Krause and Daniel Golovin. Submodular Function Maximization. In Lucas Bordeaux, Youssef Hamadi, Pushmeet Kohli, and Robert Mateescu, editors, *Tractability*, pages 71–104. Cambridge, 2013. DOI: 10.1017/CBO9781139177801.004.

[45] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012. DOI: 10.1561/2200000044.

[46] Renaud Lacour, Kathrin Klamroth, and Carlos M. Fonseca. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, 79:347–360, March 2017. DOI: 10.1016/j.cor.2016.06.021.

[47] Qiaohao Liang and Lipeng Lai. Scalable Bayesian Optimization Accelerates Process Optimization of Penicillin Production. In *NeurIPS 2021 AI for Science Workshop*, September 2021.

[48] Joseph Victor Michalowicz, Jonathan M. Nichols, and Frank Bucholtz. *Handbook of Differential Entropy*. 2013.

[49] Thomas P. Minka. Expectation Propagation for Approximate Bayesian Inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pages 362–369, San Francisco, CA, USA, August 2001.

[50] Henry B. Moss, David S. Leslie, Javier Gonzalez, and Paul Rayson. GIBBON: General-purpose Information-Based Bayesian Optimisation. *Journal of Machine Learning Research*, 22(235):1–49, 2021.

[51] Henry B. Moss, David S. Leslie, and Paul Rayson. MUMBO: MUlti-task Max-Value Bayesian Optimization. In *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 447–462, Cham, 2021. DOI: 10.1007/978-3-030-67664-3_27.

[52] Dang Nguyen, Sunil Gupta, Santu Rana, Alistair Shilton, and Svetha Venkatesh. Bayesian Optimization for Categorical and Category-Specific Continuous Inputs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:5256–5263, April 2020. DOI: 10.1609/aaai.v34i04.5971.

[53] Majid Nour, Zafer Cömert, and Kemal Polat. A Novel Medical Diagnosis model for COVID-19 infection detection based on Deep Features and Bayesian Optimization. *Applied Soft Computing*, 97:106580, December 2020. DOI: 10.1016/j.asoc.2020.106580.

[54] Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A Flexible Framework for Multi-Objective Bayesian Optimization using Random Scalarizations. In *Uncertainty in Artificial Intelligence*, pages 766–776, August 2020.

[55] Michael G. Parsons and Randall L. Scott. Formulation of Multicriterion Design Optimization Problems for Solution With Scalar Numerical Optimization Methods. *Journal of Ship Research*, 48(01):61–76, March 2004. DOI: 10.5957/jsr.2004.48.1.61.

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[57] Valerio Perrone, Iaroslav Shcherbatyi, Rodolphe Jenatton, Cedric Archambeau, and Matthias Seeger. Constrained Bayesian Optimization with Max-Value Entropy Search. *NeurIPS Workshop on Meta-Learning*, October 2019.

[58] Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, volume 20, pages 1177–1184, 2008.

[59] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass, 2006.

[60] Binxin Ru, Ahsan Alvi, Vu Nguyen, Michael A. Osborne, and Stephen Roberts. Bayesian Optimisation over Multiple Continuous and Categorical Inputs. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8276–8285, November 2020.

[61] Pratyush Sen and Jian-Bo Yang. *Multiple Criteria Decision Support in Engineering Design*. London, softcover reprint of the original 1st ed. 1998 edition edition, January 1998.

[62] Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

[63] 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, January 2016. DOI: 10.1109/JPROC.2015.2494218.

[64] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 2951–2959, 2012.

[65] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable Bayesian Optimization Using Deep Neural Networks. In *International Conference on Machine Learning*, pages 2171–2180, June 2015.

[66] Jialin Song, Yuxin Chen, and Yisong Yue. A General Framework for Multi-fidelity Bayesian Optimization with Gaussian Processes. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 3158–3167, April 2019.

[67] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise Safe Bayesian Optimization with Gaussian Processes. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4781–4789, July 2018.

[68] Shinya Suzuki, Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Multi-objective Bayesian Optimization using Pareto-frontier Entropy. In *International Conference on Machine Learning*, pages 9279–9288, November 2020.

[69] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian Optimization with Max-value Entropy Search and its Parallelization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9334–9345, November 2020.

[70] Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Sequential- and Parallel-Constrained Max-value Entropy Search via Information Lower Bound. *arXiv:2102.09788*, November 2021.

[71] Ryoji Tanabe and Hisao Ishibuchi. An Easy-to-use Real-world Multi-objective Optimization Problem Suite. *Applied Soft Computing*, 89:106078, April 2020. DOI: 10.1016/j.asoc.2020.106078.

[72] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509, September 2008. DOI: 10.1007/s10898-008-9354-2.

[73] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. DOI: 10.1038/s41592-019-0686-2.

[74] Zi Wang and Stefanie Jegelka. Max-value Entropy Search for Efficient Bayesian Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3627–3635, July 2017.

[75] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[76] James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently Sampling Functions from Gaussian Process Posteriors. In *International Conference on Machine Learning*, pages 10292–10302, November 2020.

[77] James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Pathwise Conditioning of Gaussian Processes. *Journal of Machine Learning Research*, 22(105):1–47, 2021.

[78] Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, March 2019. DOI: 10.11989/JEST.1674-862X.80904120.

[79] Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical Multi-fidelity Bayesian Optimization for Hyperparameter Tuning. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, pages 788–798, August 2020.

[80] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*, 75(1): 3–34, September 2019. DOI: 10.1007/s10898-019-00798-7.

[81] Richard Zhang and Daniel Golovin. Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization. In *International Conference on Machine Learning*, pages 11096–11105, November 2020.

[82] Yehong Zhang, Trong Nghia Hoang, Bryan Kian Hsiang Low, and Mohan Kankanhalli. Information-Based Multi-Fidelity Bayesian Optimization. *NIPS Workshop on Bayesian Optimization*, page 5, 2017.

[83] 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, April 2003. DOI: 10.1109/TEVC.2003.810758.

[84] Eckart Zitzler and Lothar Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In Agoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature — PPSN V*, Lecture Notes in Computer Science, pages 292–301, Berlin, Heidelberg, 1998. DOI: 10.1007/BFb0056872.

[85] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, June 2000. DOI: 10.1162/106365600568202.

[86] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 862–876, Berlin, Heidelberg, 2007. DOI: 10.1007/978-3-540-70928-2_64.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]

(c) Did you discuss any potential negative societal impacts of your work? [N/A]

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes] The proofs for the main results are in the appendix.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code is in the supplementary material and will be made public in due time.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The experimental set-up is described in Appendix I.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We do not include the error bars directly on the plots because it made it difficult to visually compare between the different algorithms. Alternatively, we opted to plot the different quantiles side by side e.g. Figure 16.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The computing resources are described in the caption of the wall times plots e.g. Appendix I.6.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [No] We only cited the original code and corresponding manuscript in Appendix I.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code is in the supplementary material and will be made public in due time.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

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

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

Appendix to:

Joint Entropy Search for Multi-Objective Bayesian Optimization

A Basics

A.1 Bayesian optimization pseudocode

Bayesian optimization consists of two steps that are iterated until the budget of function evaluations N is exhausted. The first step is modelling, this is where posterior of the probabilistic model $p(f|D_n)$ is computed. The second step is acquisition, this is where a utility function is optimized in order to determine the next location to query. The pseudo-code for this procedure is presented in Algorithm 2. For a more thorough overview on Bayesian optimization consult the references [11, 25, 63].

Algorithm 2: Multi-objective Bayesian Optimization.

Input : A black-box function f.

Initialize the probabilistic model $p(f)$.
for $n = 0, \dots, N-1$ **do**
  Optimize for the next point $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathbb{X}} \alpha(\mathbf{x}|D_n)$.
  Evaluate the function $\mathbf{y}_{n+1} = f(\mathbf{x}_{n+1}) + \boldsymbol{\epsilon}$.
  Augment the data set $D_{n+1} = D_n \cup \{(\mathbf{x}_{n+1}, \mathbf{y}_{n+1})\}$.
  Compute the posterior $p(f|D_n)$.
end
return D_N and $p(f|D_N)$.

A.2 Posterior Gaussian process

Under the independent Gaussian observation model described in Section 2, the posterior $p(f(X)|D_n)$ evaluated at a vector $X \subset \mathbb{X}$, is a collection of Gaussian processes [59] with mean

$$\begin{aligned}&\mu_n^{(m)}(X)\\&= \mu_0^{(m)}(X) + \Sigma_0^{(m)}(X, X_n)\left(\Sigma_0^{(m)}(X_n, X_n) + \text{diag}(\sigma^{(m)}(X_n))\right)^{-1}(Y_n^{(m)} - \mu_0^{(m)}(X_n))\end{aligned} \tag{13}$$

and covariance

$$\begin{aligned}&\Sigma_n^{(m)}(X, X)\\&= \Sigma_0^{(m)}(X, X) - \Sigma_0^{(m)}(X, X_n)\left(\Sigma_0^{(m)}(X_n, X_n) + \text{diag}(\sigma^{(m)}(X_n))\right)^{-1}\Sigma_0^{(m)}(X_n, X),\end{aligned} \tag{14}$$

where we denote the collection of sampled locations by $X_n = \{\mathbf{x}_t\}_{t=1,\dots,n}$ and observations $Y_n^{(m)} = \{y_t^{(m)}\}_{t=1,\dots,n}$ for objectives $m = 1, \dots, M$.

A.3 Sampling from a Gaussian process

Exact sampling of a one-dimensional Gaussian process over a finite set, $X \subset \mathbb{X}$, scales cubically with the number of points $|X|$. This cubic cost arises form the inversion of the covariance matrix. As we want to sample global maximizers and maximums $(\mathbb{X}^*, \mathbb{Y}^*)$, exact sampling over the whole space $\mathbb{X}$ is not computational feasible. We could restrict the sampling to a discrete subset of $\mathbb{X}$, but this would require designing a principled strategy to select a reasonable subset, for example we could try triangulating between existing points [32]. To avoid this difficulty, we follow the example of previous work [36, 74], which considers generating approximate samples via random Fourier features [58, 76, 77]. The strategy centres around approximating the covariance kernel using a feature representation $\Sigma_0^{(m)}(\mathbf{x}, \mathbf{x}') \approx \varphi(\mathbf{x})^T\varphi(\mathbf{x}')$. For a stationary covariance kernel $\Sigma_0^{(m)}$, the Fourier transform of the kernel is a non-negative measure that can be normalized to obtain a probability distribution $p(\boldsymbol{\theta})$—this result follows from Bochner's Theorem [9] . For the exponential and Matern kernels, this probability distribution is known in closed-form [10, 58]. Assuming $\Sigma_0^{(m)}$ is stationary,

an approximate sample of the Gaussian process prior $\text{GP}(\mu_0^{(m)}, \Sigma_0^{(m)})$ can be written as a Bayesian linear model

$$f_0^{(m)}(\cdot) = \mu_0^{(m)}(\cdot) + \sum_{i=1}^{L} \omega_i \varphi_i(\cdot) \tag{15}$$

where $\varphi_i(\mathbf{x}) = \sqrt{2/L}\cos(\boldsymbol{\theta}_i^T \mathbf{x} + \tau_i)$ are the Fourier features depending on the random variables $\tau_i \sim U(0, 2\pi)$ and $\boldsymbol{\theta}_i \sim p(\boldsymbol{\theta})$, whilst $\omega_i \sim \mathcal{N}(0, 1)$ are the random weights for $i = 1, \dots, L$. The posterior samples of $\text{GP}(\mu_n^{(m)}, \Sigma_n^{(m)})$ can be obtained by adding an additional pathwise update via Matherons' rule [76, 77]:

$$f_n^{(m)}(\cdot) = f_0^{(m)}(\cdot) + \sum_{t=1}^{n} \kappa_t \Sigma_0^{(m)}(\cdot, \mathbf{x}_t), \tag{16}$$

where $\kappa_t = (\Sigma_0^{(m)}(X_n, X_n) + \text{diag}(\sigma^{(m)}(X_n)))^{-1}(Y_n - f_0^{(m)}(X_n))$. For a more comprehensive overview of sampling from a Gaussian process refer to [77].

B Related work

In this work we devised a number of conditional entropy estimates for the JES acquisition function. These estimates can also be used by the MES acquisition function. In this brief section, we elaborate on the similarities of our estimates with the existing work relating to the MES acquisition function.

Firstly, the noiseless entropy estimate (9) was derived recently [68] to extend the noiseless MES acquisition function to the multi-objective setting. This extension was called the Pareto frontier entropy search, whilst we refer to this acquisition function as MES-0 for clarity. An earlier attempt to extend the MES to noiseless setting resulted in the MESMO [4] acquisition function. MESMO crudely approximates the dominated space by a single box $\mathbb{D}_{\preceq}(\mathbb{Y}^*) \approx B = (-\infty, \max_{\mathbf{y}\in\mathbb{Y}^*} y^{(1)}] \times \cdots \times (-\infty, \max_{\mathbf{y}\in\mathbb{Y}^*} y^{(M)}]$. Empirically, this approximation leads to a deterioration in performance [68].

The closest work to the lower bound entropy estimate is the GIBBON [50] acquisition function. The GIBBON acquisition function was derived as a lower bound to the multi-fidelity MES acquisition function for a single objective problem. The MES-LB2 can be interpreted as the multi-objective extension of GIBBON for the single fidelity setting.

The Monte Carlo entropy estimate has been used before for the multi-fidelity MES acquisition in the single-objective setting [51, 69]. The MES-MC estimate can be interpreted as the multi-objective extension of these approaches for the single fidelity setting.

C Proof of results

C.1 Proof of Proposition 1

The upper bound property follows from the standard result that conditioning on more variables will never increase the entropy (Theorem 2.6.5 of [13]): $H(A|B) \leq H(A)$ for random variables A and B. Using this result, $H[p(\mathbf{y}|\mathbf{x}, D_n, (\mathbb{X}^*, \mathbb{Y}^*))] \leq \max(H[p(\mathbf{y}|\mathbf{x}, D_n, \mathbb{X}^*)], H[p(\mathbf{y}|\mathbf{x}, D_n, \mathbb{Y}^*)])$. Plugging this inequality into (3), we obtain the $\alpha^{\text{JES}}(\mathbf{x}|D_n) \geq \max(\alpha^{\text{PES}}(\mathbf{x}|D_n), \alpha^{\text{MES}}(\mathbf{x}|D_n))$, which implies the result.

■

C.2 Proof of Lemma 1

Under the assumptions of Lemma 1, the following series of equations holds:

$$p(\mathbf{z} \preceq \mathbb{Y}^*) = \int_{\mathbb{D}_{\preceq}(\mathbb{Y}^*)} p(\mathbf{z}) d\mathbf{z} = \sum_{j=1}^{J} \int_{B_j} \left(\prod_{m=1}^{M} p(z^{(m)}) \right) d\mathbf{z} = \sum_{j=1}^{J} \prod_{m=1}^{M} \int_{l_j^{(m)}}^{u_j^{(m)}} p(z^{(m)}) dz^{(m)}$$
$$= \sum_{j=1}^{J} \prod_{m=1}^{M} \left[\Phi\left(\frac{u_j^{(m)} - a^{(m)}}{\sqrt{b^{(m)}}} \right) - \Phi\left(\frac{l_j^{(m)} - a^{(m)}}{\sqrt{b^{(m)}}} \right) \right].$$

The third equality follows from the box decomposition of $\mathbb{D}_{\preceq}(\mathbb{Y}^*)$ and the independence between the components of $\mathbf{z}$, whilst the last equality follows from the definition of the CDF.

■

C.3 Proof of Proposition 2

In the noiseless setting, the density of interest is

$$p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*) = \frac{p(\mathbf{y}|\mathbf{x}, D_{n*})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} \mathbb{I}[\mathbf{y} \preceq \mathbb{Y}^*]. \tag{17}$$

The corresponding entropy is

$$\begin{aligned}
&H[p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)] \\
&= -\int_{\mathbb{D}_{\preceq}(\mathbb{Y}^*)} \frac{p(\mathbf{y}|\mathbf{x}, D_{n*})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} \log\left(\frac{p(\mathbf{y}|\mathbf{x}, D_{n*})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} \right) d\mathbf{y} \\
&= -\frac{1}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} \int_{\mathbb{D}_{\preceq}(\mathbb{Y}^*)} p(\mathbf{y}|\mathbf{x}, D_{n*}) \log(p(\mathbf{y}|\mathbf{x}, D_{n*})) d\mathbf{y} + \log(p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})),
\end{aligned}$$

where the CDF is given by

$$\begin{aligned}
p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*}) &= \sum_{j=1}^{J} \prod_{m=1}^{M} \left[\Phi\left(\frac{u_j^{(m)} - \mu_{n*}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x})}} \right) - \Phi\left(\frac{l_j^{(m)} - \mu_{n*}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x})}} \right) \right] \\
&= \sum_{j=1}^{J} \prod_{m=1}^{M} \left[\Phi(\gamma_m(u_j^{(m)})) - \Phi(\gamma_m(l_j^{(m)})) \right] \\
&= \sum_{j=1}^{J} \prod_{m=1}^{M} W_{j,m} = \sum_{j=1}^{J} W_j = W
\end{aligned} \tag{18}$$

and the integral is given by

$$\begin{aligned}
&-\int_{\mathbb{D}_{\preceq}(\mathbb{Y}^*)} p(\mathbf{y}|\mathbf{x}, D_{n*}) \log(p(\mathbf{y}|\mathbf{x}, D_{n*})) d\mathbf{y} \\
&= -\sum_{j=1}^{J} \int_{B_j} \prod_{m'=1}^{M} p(y^{(m')}|\mathbf{x}, D_{n*}) \sum_{m=1}^{M} \log(p(y^{(m)}|\mathbf{x}, D_{n*})) d\mathbf{y} \\
&= -\sum_{j=1}^{J} \sum_{m=1}^{M} \int_{B_j} \prod_{m'=1}^{M} p(y^{(m')}|\mathbf{x}, D_{n*}) \log(p(y^{(m)}|\mathbf{x}, D_{n*})) d\mathbf{y} \\
&= -\sum_{j=1}^{J} \sum_{m=1}^{M} \int_{B_j} p(y^{(m)}|\mathbf{x}, D_{n*}) \log(p(y^{(m)}|\mathbf{x}, D_{n*})) \prod_{m' \neq m} p(y^{(m')}|\mathbf{x}, D_{n*}) d\mathbf{y} \\
&= -\sum_{j=1}^{J} \sum_{m=1}^{M} \frac{W_j}{W_{j,m}} \int_{l_j^{(m)}}^{u_j^{(m)}} p(y^{(m)}|\mathbf{x}, D_{n*}) \log(p(y^{(m)}|\mathbf{x}, D_{n*})) dy^{(m)}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=1}^{J}\sum_{m=1}^{M} W_j \left[-\int_{l_j^{(m)}}^{u_j^{(m)}} \frac{p(y^{(m)}|\mathbf{x}, D_{n*})}{W_{j,m}} \log\left(\frac{p(y^{(m)}|\mathbf{x}, D_{n*})}{W_{j,m}}\right) dy^{(m)} - \log(W_{j,m}) \right] \\
&= \sum_{j=1}^{J}\sum_{m=1}^{M} W_j \left[\left(\frac{1}{2}\log(2\pi e) + \frac{1}{2}\log(\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})) + \log(W_{j,m}) - \frac{1}{2}\frac{V_{j,m}}{W_{j,m}} \right) - \log(W_{j,m}) \right] \\
&= \frac{MW}{2}\log(2\pi e) + \frac{W}{2}\sum_{m=1}^{M}\log(\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})) - \frac{1}{2}\sum_{j=1}^{J} W_j \sum_{m=1}^{M}\frac{V_{j,m}}{W_{j,m}}.
\end{aligned}
$$

The first equality follows from the box decomposition and the independence between the objectives, $p(\mathbf{y}|\mathbf{x}, D_{n*}) = \prod_{m=1}^{M} p(y^{(m)}|\mathbf{x}, D_{n*})$. The fourth equality follows from the definition that $W_{j,m'} = \int_{l_j^{(m')}}^{u_j^{(m')}} p(y^{(m')}|\mathbf{x}, D_{n*}) dy^{(m')}$ and $W_j = \prod_{m=1}^{M} W_{j,m}$. The sixth equality comes from noticing that $p(y^{(m)}|\mathbf{x}, D_{n*})/W_{j,m}$ is a truncated normal distribution on $[l_j^{(m)}, u_j^{(m)}]$ whose entropy can be computed analytically (Section 4.26 of [48]). Combining all the expression above gives the final result.

■

C.4 Proof of Proposition 3

In the noisy setting, the density of interest is

$$
p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*) = \frac{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} p(\mathbf{y}|\mathbf{x}, D_{n*}),
$$

where $p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*}) = W$ from (18). To obtain a tractable expression for the cumulative distribution $p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})$, we first compute the conditional distribution $p(\mathbf{y}|\mathbf{x}, D_{n+})$. To that end, let $p(\mathbf{y}|\mathbf{x}, D_{n*}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{n*}(\mathbf{x}), \boldsymbol{\Sigma}_{n*}(\mathbf{x},\mathbf{x}))$, then by standard Gaussian conditioning $p(\mathbf{y}|\mathbf{x}, D_{n+}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{n+}(\mathbf{x})$ where

$$
\begin{aligned}
\mu_{n+}^{(m)}(\mathbf{x}) &= \mu_{n*}^{(m)}(\mathbf{x}) + \Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})(\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x}))^{-1}(y^{(m)} - \mu_{n*}^{(m)}(\mathbf{x})) \\
&= \mu_{n*}^{(m)}(\mathbf{x}) + \rho_m(\mathbf{x})\frac{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})}}(y^{(m)} - \mu_{n*}^{(m)}(\mathbf{x}))
\end{aligned}
$$

and

$$
\begin{aligned}
\Sigma_{n+}^{(m)}(\mathbf{x},\mathbf{x}) &= \Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) - \Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})\left(\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})\right)^{-1}\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})(\mathbf{x},\mathbf{x}) \\
&= \Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})(1 - \rho_m^2(\mathbf{x}))
\end{aligned}
$$

with $\rho_m := \rho_m(\mathbf{x}) = \sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}/\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})}$ denoting correlation between the observation $y^{(m)}$ and the objective value $f^{(m)}(\mathbf{x})$, for objectives $m = 1, \ldots, M$. Using the box decomposition, the posterior CDF is equal to

$$
p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+}) = \sum_{j=1}^{J}\prod_{m=1}^{M}\left[\Phi\left(\frac{u_j^{(m)} - \mu_{n+}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n+}^{(m)}(\mathbf{x},\mathbf{x})}}\right) - \Phi\left(\frac{l_j^{(m)} - \mu_{n+}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n+}^{(m)}(\mathbf{x},\mathbf{x})}}\right)\right]. \tag{19}
$$

To simplify the notation we perform a change of variable:

$$
\begin{aligned}
\gamma_m^+(z) := \frac{z - \mu_{n+}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n+}^{(m)}(\mathbf{x},\mathbf{x})}} &= \frac{z^{(m)} - \mu_{n*}^{(m)}(\mathbf{x}) - \rho_m \frac{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})}}(y^{(m)} - \mu_{n*}^{(m)}(\mathbf{x}))}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})(1 - \rho_m^2)}} \\
&= \frac{\frac{z^{(m)} - \mu_{n*}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}} - \rho_m \frac{(y^{(m)} - \mu_{n*}^{(m)}(\mathbf{x}))}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})}}}{\sqrt{1 - \rho_m^2}} = \frac{\gamma_m(z) - \rho_m \bar{y}^{(m)}}{\sqrt{1 - \rho_m^2}}
\end{aligned}
$$

where $\gamma_m(z) = (z - \mu_{n*}^{(m)}(\mathbf{x}))/\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}$ is the m-th standardized value and $\bar{y}^{(m)} = (y^{(m)} - \mu_{n*}^{(m)}(\mathbf{x}))/\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})}$ is the standardized observation for $m = 1, \ldots, M$. To derive the moments of $p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)$, we first obtain the moment generating function of the standardized observation $\bar{\mathbf{y}}$.

$$
\begin{aligned}
M_{\bar{\mathbf{y}}}(\mathbf{t}) &= \frac{1}{W}\int_{\mathbb{R}^M} e^{\mathbf{t}^T\bar{\mathbf{y}}} p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+}) p(\bar{\mathbf{y}}|\mathbf{x}, D_{n*}) d\bar{\mathbf{y}} \\
&= \frac{1}{W}\int_{\mathbb{R}^M} \prod_{m=1}^{M} e^{t^{(m)}\bar{y}^{(m)}} \left(\sum_{j=1}^{J} \prod_{m=1}^{M} \left[\Phi(\gamma_m^+(u_j^{(m)})) - \Phi(\gamma_m^+(l_j^{(m)})) \right] \right) \prod_{m=1}^{M} p(\bar{y}^{(m)}|\mathbf{x}, D_{n*}) d\bar{\mathbf{y}} \\
&= \frac{1}{W}\sum_{j=1}^{J}\prod_{m=1}^{M}\int_{\mathbb{R}} e^{t^{(m)}\bar{y}^{(m)}} \left[\Phi(\gamma_m^+(u_j^{(m)})) - \Phi(\gamma_m^+(l_j^{(m)})) \right] \phi(\bar{y}^{(m)}) d\bar{y}^{(m)} \\
&= \frac{1}{W}\sum_{j=1}^{J}\prod_{m=1}^{M} e^{\frac{(t^{(m)})^2}{2}} \int_{\mathbb{R}} \left[\Phi(\gamma_m^+(u_j^{(m)})) - \Phi(\gamma_m^+(l_j^{(m)})) \right] \phi(\bar{y}^{(m)} - t^{(m)}) d\bar{y}^{(m)}.
\end{aligned}
$$

By Lemma 2 in [2], the expectation of the normal CDF is given by $\int_{\mathbb{R}} \Phi(az + b)\phi(z)dz = \Phi(b/\sqrt{1+a^2})$ for any constants $a, b \in \mathbb{R}$. Using this result,

$$
\begin{aligned}
&\int_{\mathbb{R}} \left[\Phi(\gamma_m^+(u_j^{(m)})) - \Phi(\gamma_m^+(l_j^{(m)})) \right] \phi(\bar{y}^{(m)} - t^{(m)}) d\bar{y}^{(m)} \\
&= \int_{\mathbb{R}} \left[\Phi\left(\frac{\gamma_m(u_j^{(m)}) - \rho_m \bar{y}^{(m)}}{\sqrt{1-\rho_m^2}} \right) - \Phi\left(\frac{\gamma_m(l_j^{(m)}) - \rho_m \bar{y}^{(m)}}{\sqrt{1-\rho_m^2}} \right) \right] \phi(\bar{y}^{(m)} - t^{(m)}) d\bar{y}^{(m)} \\
&= \int_{\mathbb{R}} \left[\Phi\left(\frac{\gamma_m(u_j^{(m)}) - \rho_m (\bar{y}^{(m)} + t^{(m)})}{\sqrt{1-\rho_m^2}} \right) - \Phi\left(\frac{\gamma_m(l_j^{(m)}) - \rho_m (\bar{y}^{(m)} + t^{(m)})}{\sqrt{1-\rho_m^2}} \right) \right] \phi(\bar{y}^{(m)}) d\bar{y}^{(m)} \\
&= \Phi(\gamma_m(u_j^{(m)}) - \rho_m t^{(m)}) - \Phi(\gamma_m(l_j^{(m)}) - \rho_m t^{(m)}).
\end{aligned}
$$

The first moment $p(\bar{y}^{(m)}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)$ can be obtained by evaluating the first derivative of $M_{\bar{\mathbf{y}}}(\mathbf{t})$ with respect to $t^{(m)}$ at $\mathbf{t} = \mathbf{0}_M$.

$$
\begin{aligned}
&\mathbb{E}[\bar{y}^{(m)}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*] = \frac{\partial}{\partial t^{(m)}} M_{\bar{\mathbf{y}}}(\mathbf{t})\Big|_{\mathbf{t}=\mathbf{0}_M} \\
&= \frac{1}{W}\sum_{j=1}^{J}\prod_{m' \neq m} e^{\frac{(t^{(m')})^2}{2}} \left[\beta_{j,m'}^u - \beta_{j,m'}^l \right] e^{\frac{(t^{(m)})^2}{2}} \left(t^{(m)}[\beta_{j,m}^u - \beta_{j,m}^l] - \rho_m[\zeta_{j,m}^u - \zeta_{j,m}^l] \right)\Big|_{\mathbf{t}=\mathbf{0}_M} \\
&= \frac{1}{W}\sum_{j=1}^{J}\prod_{m' \neq m} \left[\Phi(\gamma_{m'}(u_j^{(m')})) - \Phi(\gamma_{m'}(l_j^{(m')})) \right] \left(-\rho_m \left(\phi(\gamma_m(u_j^{(m)})) - \phi(\gamma_m(l_j^{(m)})) \right) \right) \\
&= -\frac{\rho_m}{W}\sum_{j=1}^{J}\frac{W_j}{W_{j,m}} G_{j,m}
\end{aligned}
$$

where for shorthand we defined the intermediate variables $\beta_{j,m}^u = \Phi(\gamma_m(u_j^{(m)}) - \rho_m t^{(m)})$, $\beta_{j,m}^l = \Phi(\gamma_m(l_j^{(m)}) - \rho_m t^{(m)})$, $\zeta_{j,m}^u = \phi(\gamma_m(u_j^{(m)}) - \rho_m t^{(m)})$ and $\zeta_{j,m}^l = \phi(\gamma_m(l_j^{(m)}) - \rho_m t^{(m)})$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$. Differentiating a second time, we can obtain the second moments. For $m \neq m'$,

$$
\begin{aligned}
\mathbb{E}[\bar{y}^{(m)}\bar{y}^{(m')}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*] &= \frac{\partial}{\partial t^{(m)}\partial t^{(m')}} M_{\bar{\mathbf{y}}}(\mathbf{t})\Big|_{\mathbf{t}=\mathbf{0}_M} \\
&= \frac{\rho_m \rho_{m'}}{W}\sum_{j=1}^{J}\frac{W_j}{W_{j,m}W_{j,m'}} G_{j,m}G_{j,m'}
\end{aligned}
$$

and for $m = m'$,

$$
\begin{aligned}
&\mathbb{E}[(\bar{y}^{(m)})^2|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*] = \frac{\partial^2}{(\partial t^{(m)})^2} M_{\bar{\mathbf{y}}}(\mathbf{t})\Big|_{\mathbf{t}=\mathbf{0}_M} \\
&= \frac{1}{W}\sum_{j=1}^{J}\prod_{m'\neq m} e^{\frac{(t^{(m')})^2}{2}}\left[\beta^u_{j,m'} - \beta^l_{j,m'}\right] e^{\frac{(t^{(m)})^2}{2}}\Big((1+(t^{(m)})^2)[\beta^u_{j,m} - \beta^l_{j,m}] \\
&\qquad -2\rho_m t^{(m)}[\zeta^u_{j,m} - \zeta^l_{j,m}] - \rho_m^2[\tau^u_{j,m}\zeta^u_{j,m} - \tau^l_{j,m}\zeta^l_{j,m}]\Big)\Big|_{\mathbf{t}=\mathbf{0}_M} \\
&= 1 + \frac{1}{W}\sum_{j=1}^{J}\prod_{m'\neq m}\left[\Phi(\gamma_{m'}(u_j^{(m')})) - \Phi(\gamma_{m'}(l_j^{(m')}))\right] \\
&\qquad \times\left(-\rho_m^2\left(\gamma_m(u_j^{(m)})\phi(\gamma_m(u_j^{(m)})) - \gamma_m(l_j^{(m)})\phi(\gamma_m(l_j^{(m)}))\right)\right) \\
&= 1 - \frac{\rho_m^2}{W}\sum_{j=1}^{J}\frac{W_j}{W_{j,m}}V_{j,m}
\end{aligned}
$$

where for shorthand we defined the intermediate variables $\tau^u_{j,m} = \gamma_m(u_j^{(m)}) - \rho_m t^{(m)}$ and $\tau^l_{j,m} = \gamma_m(l_j^{(m)}) - \rho_m t^{(m)}$ for $j = 1, \ldots, J$ and $m = 1, \ldots, M$. The moments of $y^{(m)}$ can be now derived by reversing the initial linear transformation: $y^{(m)} = \bar{y}^{(m)}\sqrt{\Sigma^{(m)}_{n*}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})} + \mu^{(m)}_{n*}(\mathbf{x})$. The first moment is

$$
\mathbb{E}[y^{(m)}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*] = \mu^{(m)}_{n*}(\mathbf{x}) - \frac{\sqrt{\Sigma^{(m)}_{n*}(\mathbf{x},\mathbf{x})}}{W}\sum_{j=1}^{J} W_j \frac{W_{j,m}}{G_{j,m}}.
$$

The second moment:

$$
\begin{aligned}
&\mathbb{E}[\bar{y}^{(m)}\bar{y}^{(m')}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*] \\
&= \mathbb{E}\left[\frac{y^{(m)} - \mu^{(m)}_{n*}(\mathbf{x})}{\sqrt{\Sigma^{(m)}_{n*}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})}}\frac{y^{(m')} - \mu^{(m')}_{n*}(\mathbf{x})}{\sqrt{\Sigma^{(m')}_{n*}(\mathbf{x},\mathbf{x}) + \sigma^{(m')}(\mathbf{x})}}\Big|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right] \\
&= \mathbb{E}\left[\frac{y^{(m)}y^{(m')} - y^{(m)}\mu^{(m')}_{n*}(\mathbf{x}) - y^{(m')}\mu^{(m)}_{n*}(\mathbf{x}) + \mu^{(m)}_{n*}(\mathbf{x})\mu^{(m')}_{n*}(\mathbf{x})}{\sqrt{\Sigma^{(m)}_{n*}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})}\sqrt{\Sigma^{(m')}_{n*}(\mathbf{x},\mathbf{x}) + \sigma^{(m')}(\mathbf{x})}}\Big|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right].
\end{aligned}
$$

This implies

$$
\begin{aligned}
&\mathbb{E}\left[y^{(m)}y^{(m')}\Big|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right] \\
&= \sqrt{\Sigma^{(m)}_{n*}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})}\sqrt{\Sigma^{(m')}_{n*}(\mathbf{x},\mathbf{x}) + \sigma^{(m')}(\mathbf{x})}\mathbb{E}[\bar{y}^{(m)}\bar{y}^{(m')}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*] \\
&\quad + \mathbb{E}[\bar{y}^{(m)}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*]\mu^{(m')}_{n*}(\mathbf{x}) + \mathbb{E}[\bar{y}^{(m')}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*]\mu^{(m)}_{n*}(\mathbf{x}) \\
&\quad - \mu^{(m)}_{n*}(\mathbf{x})\mu^{(m')}_{n*}(\mathbf{x}).
\end{aligned}
$$

Substituting in the expressions before, we have that for $m \neq m'$,

$$
\begin{aligned}
&\mathbb{E}\left[y^{(m)}y^{(m')}\Big|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right] = \mu^{(m)}_{n*}(\mathbf{x})\mu^{(m')}_{n*}(\mathbf{x}) + \frac{\sqrt{\Sigma^{(m)}_{n*}(\mathbf{x},\mathbf{x})}\sqrt{\Sigma^{(m')}_{n*}(\mathbf{x},\mathbf{x})}}{W} \\
&\quad \times\sum_{j=1}^{J} W_j\left(\frac{G_{j,m}}{W_{j,m}}\frac{G_{j,m'}}{W_{j,m'}} - \frac{\mu^{(m')}_{n*}(\mathbf{x})}{\sqrt{\Sigma^{(m')}_{n*}(\mathbf{x},\mathbf{x})}}\frac{G_{j,m}}{W_{j,m}} - \frac{\mu^{(m)}_{n*}(\mathbf{x})}{\sqrt{\Sigma^{(m)}_{n*}(\mathbf{x},\mathbf{x})}}\frac{G_{j,m'}}{W_{j,m'}}\right)
\end{aligned}
$$

and for $m = m'$

$$\mathbb{E}\left[(y^{(m)})^2 \middle| \mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right] = \mu_{n*}^{(m)}(\mathbf{x})^2 + \Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x}) - \frac{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}{W}\sum_{j=1}^{J} W_j \left(\frac{V_{j,m}}{W_{j,m}} + 2\frac{\mu_{n*}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}}\frac{G_{j,m}}{W_{j,m}}\right).$$

By some additional algebraic manipulation, the covariance for $m \neq m'$ is given

$$\mathrm{Cov}\left(y^{(m)}, y^{(m')} \middle| \mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right) = \frac{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}\sqrt{\Sigma_{n*}^{(m')}(\mathbf{x},\mathbf{x})}}{W} \times \left(\sum_{j=1}^{J} W_j \frac{G_{j,m}}{W_{j,m}}\frac{G_{j,m'}}{W_{j,m'}} - \frac{1}{W}\left(\sum_{j=1}^{J} W_j \frac{G_{j,m}}{W_{j,m}}\right)\left(\sum_{j'=1}^{J} W_{j'} \frac{G_{j',m'}}{W_{j',m'}}\right)\right)$$

and the variance is

$$\mathbb{V}\mathrm{ar}\left(y^{(m)} \middle| \mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right) = \Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x}) - \frac{\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x})}{W}\left(\sum_{j=1}^{J} W_j \frac{V_{j,m}}{W_{j,m}} + \frac{1}{W}\left(\sum_{j=1}^{J} W_j \frac{G_{j,m}}{W_{j,m}}\right)^2\right).$$

■

C.5 Proof of Proposition 4

We will prove the general statement that the mutual information is invariant under smooth bijective transformations. Consider two random variables X and Y then following equations hold:

$$I(X;Y) = \mathbb{E}_{p(X,Y)}\left[\log \frac{p(X,Y)}{p(X)p(Y)}\right] = \mathbb{E}_{p(X',Y')}\left[\log \frac{p(X',Y')|J_x||J_y|}{p(X')|J_x|p(Y')|J_y|}\right] = I(X';Y'),$$

where $X' = g_x(X)$ and $Y' = g_y(Y)$ represents the transformed variables under some suitably defined smooth invertible functions g_x and g_y. The expressions J_x and J_y correspond to the Jacobian of the transformation, which are non-zero because the transformation are assumed to be invertible. The result follows by restricting the class of bijective functions to ones where the Pareto set remains unchanged. For example, the class of monotonic increasing functions in each objective ensures this property holds: $g : \mathbb{R}^M \to \mathbb{R}^M$ such that $g(\mathbf{y}) = (g_1(y^{(1)}), \dots g_M(y^{(M)}))$ with g_m being monotonically increasing.

■

D Monte Carlo entropy estimate

In this section, we consider estimating the entropy of $p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)$ via Monte Carlo. As a reminder, the entropy of interest can be written as an expectation over $p(\mathbf{y}|\mathbf{x}, D_{n*})$:

$$\begin{aligned}
&H[p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)] \\
&= -\int_{\mathbb{R}^M} p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*) \log\left(p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)\right) d\mathbf{y} \\
&= -\int_{\mathbb{R}^M} \frac{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} p(\mathbf{y}|\mathbf{x}, D_{n*}) \log\left(\frac{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} p(\mathbf{y}|\mathbf{x}, D_{n*})\right) d\mathbf{y} \\
&= -\mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}\left[\frac{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} \log\left(\frac{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})} p(\mathbf{y}|\mathbf{x}, D_{n*})\right)\right].
\end{aligned}$$

The CDF in the denominator is $p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*}) = W$ from (18), whilst the CDF in the numerator is

$$p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+}) = \sum_{j=1}^{J} \prod_{m=1}^{M} \left[\Phi\left(\gamma_m^+(u_j^{(m)})\right) - \Phi\left(\gamma_m^+(l_j^{(m)})\right)\right]$$
$$= \sum_{j=1}^{J} \prod_{m=1}^{M} W_{j,m}^+(\mathbf{y}) = \sum_{j=1}^{J} W_j^+(\mathbf{y}) = W^+(\mathbf{y})$$

from (19). By sampling $\mathbf{y}_i \sim p(\mathbf{y}|\mathbf{x}, D_{n*})$ for $i = 1, \ldots, I$, we can approximate the entropy with the following Monte Carlo average:

$$H[p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)] = -\frac{1}{W}\mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}\left[W^+(\mathbf{y})\log(W^+(\mathbf{y})p(\mathbf{y}|\mathbf{x}, D_{n*}))\right] + \log(W)$$
$$\approx h^{\text{JES-MC}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n),$$

where the Monte Carlo entropy estimate is given by

$$h^{\text{JES-MC}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n) = -\frac{1}{WI}\sum_{i=1}^{I} W^+(\mathbf{y}_i)\log(W^+(\mathbf{y}_i)p(\mathbf{y}_i|\mathbf{x}, D_{n*})) + \log(W). \quad (20)$$

Instead of generating new samples for each call of the acquisition function, we follow the general wisdom in BO [75] and apply the reparametrization trick on the sampling distribution: $\mathbf{y}_i = \boldsymbol{\mu}_{n*}(\mathbf{x}) + \mathbf{C}_{n*}(\mathbf{x})\mathbf{z}_i$, where $\mathbf{C}_{n*}(\mathbf{x}) \in \mathbb{R}^{M\times M}$ is the Cholesky factor of $\boldsymbol{\Sigma}_{n*}(\mathbf{x}, \mathbf{x})$ and $\mathbf{z}_i \sim \mathcal{N}(0, I_M)$ are the base samples that only need to be initialized once. The variance of this estimate could potentially be reduced by including control variates. For example,

$$h^{\text{JES-MC-CV}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n) = h^{\text{JES-MC}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n)$$
$$+ \beta_1\left(\frac{1}{I}\sum_{i=1}^{I} W^+(\mathbf{y}_i) - \mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}[W^+(\mathbf{y})]\right)$$
$$+ \beta_2\left(-\frac{1}{I}\sum_{i=1}^{I}\log(p(\mathbf{y}_i|\mathbf{x}, D_{n*})) + \mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}[\log(p(\mathbf{y}|\mathbf{x}, D_{n*})]\right),$$

where the expectations are known,

$$\mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}[W^+(\mathbf{y})] = W, \quad (21)$$

$$-\mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}[\log(p(\mathbf{y}|\mathbf{x}, D_{n*})] = \frac{M}{2}\log(2\pi e) + \frac{1}{2}\sum_{m=1}^{M}\log(\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x}) + \sigma^{(m)}(\mathbf{x})) \quad (22)$$

and $\beta_1, \beta_2 \in \mathbb{R}$ are the regression coefficients. We did not assess the effect adding control variates to our Monte Carlo estimate because the standard quasi-Monte Carlo scheme with the reparametrization trick worked reasonably well out of the box. One word of caution [39]: combining both control variates and quasi-Monte Carlo could lead to an increase in variance if the coefficients are naively estimated.

E Submodularity

A set function $g : 2^V \to \mathbb{R}$ is submodular if for every $A, B \subset V$, $g(A\cap B)+g(A\cup B) \leq g(A)+g(B)$, where V is the set of interest—for further details on submodularity refer to [44]. The following proposition states that the lower bound batch acquisition function and its approximation for the JES and MES are submodular functions defined over subsets of the input space.

Proposition 5 *The lower bound batch acquisition function $\alpha^{\text{qLB-JES}}(X|D_n)$ and its approximations $\hat{\alpha}^{\text{qLB-JES}}(X|D_n)$ are submodular functions defined over subsets of $\mathbb{X}$.*

Proof. Let $X = \{\mathbf{x}_i\}_{i=1,\dots,q} \subset \mathbb{X}$ denote a set of inputs. If $K_X \in \mathbb{R}^{q\times q}$ is a positive semi-definite matrix such that $K_{i,j}$ depends only on the inputs $\mathbf{x}_i \in \mathbb{X}$ and $\mathbf{x}_j \in \mathbb{X}$, then the log-determinant function $\log\det K_X$ defined over sets X is submodular [45]. Similar to [50], we will show that this lower bound batch acquisition function can be written as a sum of these log-determinants.

$$
\begin{aligned}
&\alpha^{q\text{LB-JES}}(\mathbf{x}^{[1:q]}|D_n) \\
&= \frac{1}{2}\sum_{m=1}^{M} \log\det(\Sigma_n^{(m)}(\mathbf{x}^{[1:q]},\mathbf{x}^{[1:q]}) + \text{diag}(\sigma^{(m)}(\mathbf{x}^{[1:q]}))) \\
&\quad + \frac{M}{2}\log(2\pi e) - \sum_{i=1}^{q} \mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}\left[H[p(\mathbf{y}^{[i]}|\mathbf{x}^{[i]}, D_{n*}, f(\mathbb{X}^*) \preceq \mathbb{Y}^*)]\right] \\
&= \frac{1}{2}\sum_{m=1}^{M} \log\det(\Sigma_n^{(m)}(\mathbf{x}^{[1:q]},\mathbf{x}^{[1:q]}) + \text{diag}(\sigma^{(m)}(\mathbf{x}^{[1:q]}))) + \sum_{i=1}^{q}\log\left(e^{\zeta(\mathbf{x}^{[i]})}\right) \\
&= \frac{1}{2}\sum_{m=1}^{M}\left(\log\det(\Sigma_n^{(m)}(\mathbf{x}^{[1:q]},\mathbf{x}^{[1:q]}) + \text{diag}(\sigma^{(m)}(\mathbf{x}^{[1:q]}))) + \sum_{i=1}^{q}\log\left(e^{2\zeta(\mathbf{x}^{[i]})/M}\right)\right) \\
&= \frac{1}{2}\sum_{m=1}^{M}\left(\log\det(\Sigma_n^{(m)}(\mathbf{x}^{[1:q]},\mathbf{x}^{[1:q]}) + \text{diag}(\sigma^{(m)}(\mathbf{x}^{[1:q]}))) + \log\det\left(\text{diag}\left(e^{2\zeta(\mathbf{x}^{[1:q]})/M}\right)\right)\right)
\end{aligned}
$$

where

$$
\zeta(\mathbf{x}^{[i]}) = \frac{M}{2q}\log(2\pi e) - \mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}\left[H[p(\mathbf{y}^{[i]}|\mathbf{x}^{[i]}, D_{n*}, f(\mathbb{X}^*) \preceq \mathbb{Y}^*)]\right]. \tag{23}
$$

For notational convenience, let $A^{(m)} = \Sigma_n^{(m)}(\mathbf{x}^{[1:q]},\mathbf{x}^{[1:q]}) + \text{diag}(\sigma^{(m)}(\mathbf{x}^{[1:q]}))$ for $m = 1,\dots,M$ and $B = \text{diag}\left(e^{\zeta(\mathbf{x}^{[1:q]})/M}\right)$. Combining the terms, we obtain

$$
\alpha^{q\text{LB-JES}}(\mathbf{x}^{[1:q]}|D_n) = \frac{1}{2}\sum_{m=1}^{M}\left(\log\det(A^{(m)}) + \log\det(B^2)\right) = \frac{1}{2}\sum_{m=1}^{M}\log\det K^{(m)},
$$

where $K^{(m)} = BA^{(m)}B$ with $K_{i,j}^{(m)} = e^{\zeta(\mathbf{x}^{[i]})/M}e^{\zeta(\mathbf{x}^{[j]})/M}A_{i,j}^{(m)}$ for $i,j = 1,\dots,q$ and $m = 1,\dots,M$. Each summand, $\log\det K^{(m)}$, is a submodular function [45]. As a sum of submodular functions is submodular, we conclude using that the lower bound batch acquisition function is submodular. For the approximate batch acquisition function, the expectation in (23) is replaced by the corresponding Monte Carlo estimate. The submodularity derivation above continues to hold when using the conditional entropy approximations.

■

F Cost analysis

In this section we consider the costs involved in evaluating the JES acquisition function according to Algorithm 1. We also include some discussion of the cost of the other information-theoretic criterion, namely MES and PES.

Sampling cost. The cost of approximate sampling from a single Gaussian processes $p(f^{(m)}|D_n)$ using the random Fourier features (described briefly in Appendix A.3) is $O(\min(n, L)^3)$, where L is the number of Fourier features—for more details refer to [76]. An evaluation of a sample at a set of inputs $X \subset \mathbb{X}$ has a linear cost depending on $|X|$ [76].

Multi-objective optimization cost. Given a sample f_s, we optimize for p Pareto optimal points $(\mathbb{X}_s^*, \mathbb{Y}_s^*)$ using the popular genetic algorithm known as NSGA2 [18]. The cost of this algorithm is $O(MN_{\text{pop}}^2 N_{\text{gen}} N_{\text{off}})$, where N_{pop} is the size of the population, N_{gen} is the number of generations and N_{off} is the number of offspring[18]. At a high level, NSGA2 works by evaluating the function at N_{pop} locations at time $t = 1$ and then moves on to evaluate N_{off} candidates for the rest of the time $t = 2,\dots,N_{\text{gen}}$. The location of the offspring evaluations are determined by a random heuristic motivated by the mechanisms involved in the theory of evolution: crossover, mutation, elitism and diversity.

Box decomposition. The cost of performing a single the box decomposition based on the incremental algorithm in [46] is $O(p^{\lfloor M/2 \rfloor + 1})$, where p is the number of Pareto optimal points.

Conditioning cost. Conditioning on an additional p data-points requires updating the Cholesky decomposition of the input covariance matrix. The cost of a single update of this kind relies on a triangular solve, which has a quadratic complexity $O(M(n+p)^2)$.

Initial entropy evaluation cost. The initial entropy (4) can be computed directly from the posterior covariance. The cost of instantiating the caches of the Gaussian process covariance, namely $(\Sigma_0^{(m)}(X_n, X_n) + \text{diag}(\sigma^{(m)}(X_n))^{-1}$ is $O(n^3)$. The cost of evaluating the posterior covariance (14) at point $\mathbf{x} \in \mathbb{X}$ is $O(n^2)$. In the batch case, we also have to compute a log-determinant of the $q \times q$ covariance matrix, which has a cost of $O(q^3)$. We use the implementation in the GPyTorch [26], which computes the approximate log-determinant, which only uses a linear cost of $O(q)$—this approximation approach is outlined in [21].

Conditional entropy evaluation cost. Assuming the posterior model and conditioned model are instantiated, we will consider the operations involved when evaluating the conditional entropy estimates. The cost of evaluating the probability density function and CDF of a univariate normal distribution at a point $\mathbf{x} \in \mathbb{X}$ are both assumed to be $O(1)$. As a result, the dominant cost of evaluating the $h^{\text{JES-0}}$ (9) and $h^{\text{JES-LB2}}$ (11) comes from the evaluation of the variance. For $h^{\text{JES-LB}}$ (10), we additionally have to populate an $M \times M$ matrix and compute its log-determinant. For $h^{\text{JES-MC}}$ (20), we need to initialize a set of I base samples for the reparametrization trick [75] described in Appendix D—this has a linear one time cost of $O(MI)$. Assuming the Monte Carlo samples are generated, the rest of the operations depend linear on I.

Expectation propagation. To approximate the density $p(\mathbf{y}|\mathbf{x}, D_n, \mathbb{X}^*)$ in the PES acquisition function (1), the authors of [27] consider using expectation propagation [49]. The dominant cost of this is from matrix inversions of the covariance matrix, which scales cubically with the number of data points in consideration. In the initialization phase, we prepare the expectation propagation caches, which require inverting an $(n+p) \times (n+p)$ matrix, whilst during testing we invert a $(q+p) \times (q+p)$ matrix, where q is the batch size.

Total cost. The total cost is the sum of the initialization cost and the query cost. Using the variables defined in Table 1 and Table 2, we can write down the initialization and query cost of JES, MES and PES. For the calculations, we assume that we have already queried n points and we want to compute the acquisition function at a batch of q points using S Monte Carlo samples of the Pareto set and/or front comprised of p points.

- **JES.** The initialization cost is $C_{\text{init}} + SC_{\text{sample}}(L) + SC_{\text{moo}}(N_{\text{pop}}, N_{\text{gen}}, N_{\text{off}}) + SC_{\text{bd}}(p) + SC_{\text{cond}}(p)$. There is an additional initial cost of $SC_{\text{base-sample}}(I)$ for the Monte Carlo conditional entropy estimate. The query cost is $C_{\text{init-h}}(q) + SqC_{\text{h}}(p)$, where $C_{\text{h}} \in \{C_{\text{h-0}}, C_{\text{h-lb}}, C_{\text{h-lb2}}, C_{\text{h-mc}}\}$ is the conditional entropy estimation strategy. Keeping the other parameters fixed, the dominant query cost is Mq^2 when using the approximate log-determinant and Mq^3 when using the standard log-determinant for large q.
- **MES.** We consider the MES algorithm obtained by excluding the conditioning step described in Algorithm 1. The initialization cost of this approach is $C_{\text{init}}(q) + SC_{\text{sample}}(L) + SC_{\text{moo}}(N_{\text{pop}}, N_{\text{gen}}, N_{\text{off}}) + SC_{\text{bd}}(p)$. The query cost is $C_{\text{init-h}}(q) + SqC_{\text{h}}(0)$, where $C_{\text{h}} \in \{C_{\text{h-0}}, C_{\text{h-lb}}, C_{\text{h-lb2}}, C_{\text{h-mc}}\}$ is the conditional entropy estimation strategy. Keeping the other parameters fixed, the dominant query cost is Mq^2 when using the approximate log-determinant and Mq^3 when using the standard log-determinant for large q.
- **PES.** We consider the expectation propagation approach described in [27] to approximate the PES acquisition. The batch extension is derived in a follow-up work [29]. The initialization cost of this approach is $C_{\text{init}}(q) + SC_{\text{sample}}(L) + SC_{\text{moo}}(N_{\text{pop}}, N_{\text{gen}}, N_{\text{off}}) + SC_{\text{ep0}}(p)$. The query cost is $C_{\text{init-h}}(q) + SC_{\text{ep}}(q, p)$. Keeping the other parameters fixed, the dominant query cost is SMq^3 for large q.

Naturally, the initialization phase can be executed in parallel because we are running S independent operations. For our experiments, we did not take advantage of this advantageous property when performing the sampling, multi-objective optimization and box decomposition.

Operation	Reference	Cost
Initializing the posterior $p(f\|D_n)$	C_{init}	Mn^3
Log-determinant of a $K \times K$ matrix	$C_{\text{logdet}}(K)$	K^3
Approximate log-determinant of a $K \times K$ matrix	$C_{\text{alogdet}}(K)$	K
Generating I base samples from $\mathcal{N}(0, \text{diag}(\mathbf{1}_M))$	$C_{\text{base-sample}}(I)$	MI
Approximate sampling of $p(f\|D_n)$	$C_{\text{sample}}(L)$	$M \min(n, L)^3$
Multi-objective optimization of f_s	$C_{\text{moo}}(N_{\text{pop}}, N_{\text{gen}}, N_{\text{off}})$	$MN_{\text{pop}}^2 N_{\text{gen}} N_{\text{off}}$
Box decomposition of $\mathbb{Y}_s^*$ with p points	$C_{\text{bd}}(p)$	$p^{\lfloor M/2 \rfloor + 1}$
Conditioning on the Pareto optimal point $p(f\|D_{n*})$	$C_{\text{cond}}(p)$	$M(n+p)^2$
Initialization of expectation propagation caches	$C_{\text{ep0}}(p)$	$M(n+p)^3$

Table 1: The initialization and operation costs. The cost only includes the highest order terms and we have ignored the constant factors.

Operation	Reference	Cost
Posterior covariance $\mathbf{\Sigma}_n(X, X)$	$C_{\text{cov}}(\|X\|, n)$	$M(\|X\|n^2 + \|X\|^2)$
Initial entropy $H[p(\mathbf{y}^{[1:q]}\|\mathbf{x}^{[1:q]}, D_n)]$	$C_{\text{init-h}}(q)$	$C_{\text{cov}}(q, n) + MC_{\text{logdet}}(q)$
Conditional entropy estimate $h^{\text{JES-0}}$	$C_{\text{h-0}}(p)$	$C_{\text{cov}}(1, n+p)$
Conditional entropy estimate $h^{\text{JES-LB}}$	$C_{\text{h-lb}}(p)$	$C_{\text{cov}}(1, n+p) + M^2 + C_{\text{logdet}}(M)$
Conditional entropy estimate $h^{\text{JES-LB2}}$	$C_{\text{h-lb2}}(p)$	$C_{\text{cov}}(1, n+p)$
Conditional entropy estimate $h^{\text{JES-MC}}$	$C_{\text{h-mc}}(p)$	$C_{\text{cov}}(1, n+p) + MI$
Expectation propagation	$C_{\text{ep}}(q, p)$	$M(q+p)^3$

Table 2: The cost of involved with querying after the initialization. The cost only includes the highest order terms and we have ignored the constant factors. In our experiments, we use the approximate log-determinant, which cost C_{alogdet} instead of the more expensive cost of C_{logdet}.

G Contour plots

In this section we present some contour plots to illustrate visually the differences that emerges between the different approximation strategies for the information-theoretic acquisition functions on a single objective problem. The approximation for PES, MES and JES are presented in Figure 7, Figure 8 and Figure 9, respectively. To obtain the ground truth for the acquisition function we run the rejection sampling scheme discussed in [36]. As a comparison we also include the contours of some popular single objective acquisition functions. Visually the estimates all appear to be reasonably effective for this set-up. Interestingly, the PES and MES prefer querying around the lower mode, whilst JES prefers the upper mode which perhaps offers a better trade-off between the gain in information about both the optimal input and output.

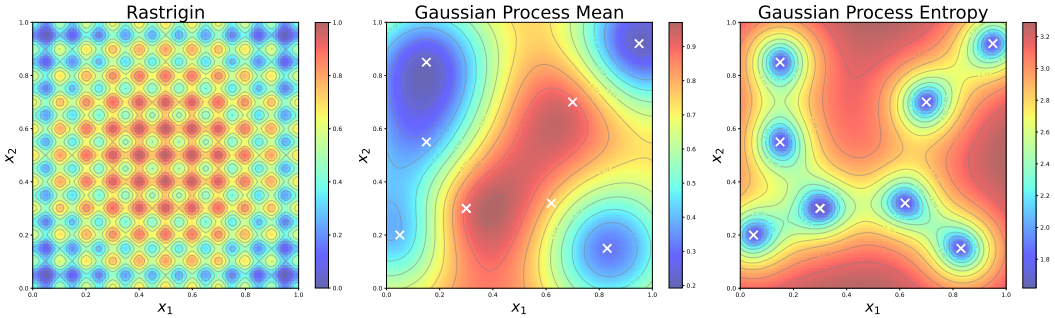


Figure 6: The contours for the Gaussian process posterior mean and entropy after making 8 noisy observations of the (normalized) Rastrigin objective function ($d = 2, M = 1$).

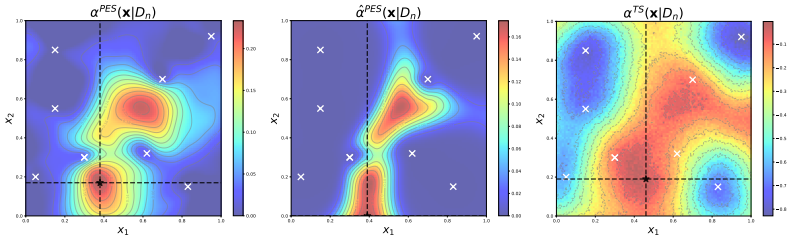


Figure 7: The contours for the predictive entropy search acquisition and its approximation obtained via expectation propagation. For reference we also include a random Thompson sample. The location of the maximizer is highlighted using a pair of dotted black lines.

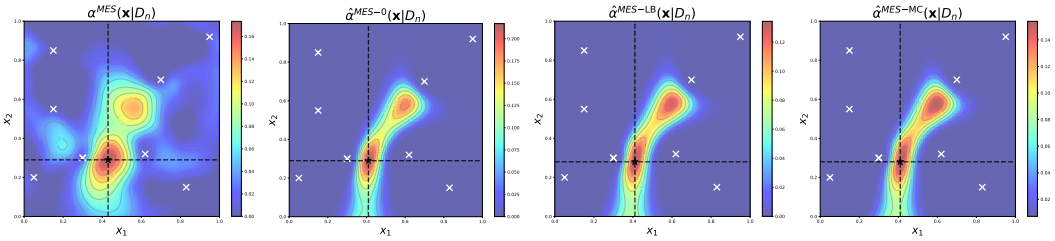


Figure 8: The contours for the maximum value entropy search acquisition function and its approximation obtained via the zero noise approximation, moment matching and Monte Carlo. The location of the maximizer is highlighted using a pair of dotted black lines.

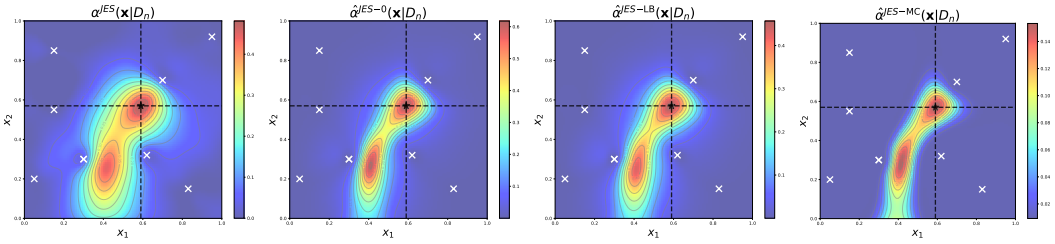


Figure 9: The contours for the joint entropy search acquisition function and its approximation obtained via the zero noise approximation, moment matching and Monte Carlo. The location of the maximizer is highlighted using a pair of dotted black lines.

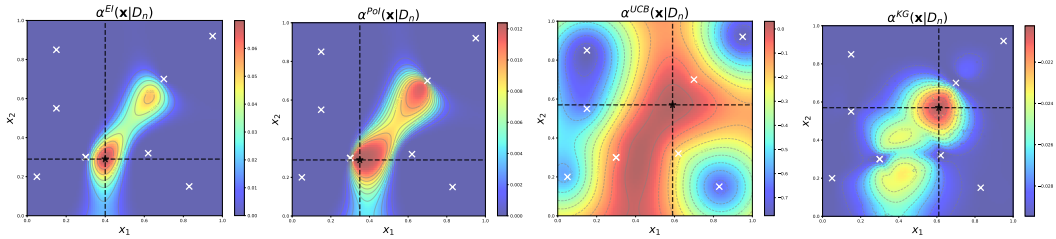


Figure 10: The contours for the expected improvement, probability of improvement, upper confidence bound ($\beta = 3$) and knowledge gradient acquisition function. The location of the maximizer is highlighted using a pair of dotted black lines.

H Hypervolume indicator

The hypervolume indicator is defined as the area of the dominated region between a reference point $\mathbf{r} \in \mathbb{R}^M$ and the set of interest $A \subset \mathbb{R}^M$:

$$U_{\text{HV}}(A) = \int_{\mathbb{R}^M} \mathbb{I}[\mathbf{r} \preceq \mathbf{z} \preceq A] d\mathbf{z}.$$

As discussed in the main text (Section 4), the HV indicator is sensitive to the parametrization of the objective space. In order to build a more complete picture about the performance of a multi-objective optimization algorithm, we consider tracking the HV under a number of different parameterization. Instead of deriving a family of transformation function, we consider a more general approach based on a different formulation of the HV indicator described in [19, 81]. In particular, the HV indicator can be written as an expectation over a probability distribution by performing a change of variables into spherical polar coordinates:

$$U_{\text{HV}}(A) = \frac{\pi^{M/2}}{2^M \Gamma(M/2+1)} \mathbb{E}_{p(\boldsymbol{\lambda})} \left[\max_{\mathbf{a} \in A} s_{\boldsymbol{\lambda}}(\mathbf{a}) \right] \tag{24}$$

where $\Gamma(\cdot)$ is the Gamma function and the scalarization function $s_{\boldsymbol{\lambda}} : \mathbb{R}^M \to \mathbb{R}$ is defined as

$$s_{\boldsymbol{\lambda}}(\mathbf{a}) = \min_{m=1,\dots,M} \left(\max \left(0, (a^{(m)} - r^{(m)})/\lambda^{(m)} \right) \right)^M. \tag{25}$$

For the standard HV, the inverse weight distribution $p(\boldsymbol{\lambda})$ is a uniform distribution over the surface of the M-dimensional unit sphere in the non-negative orthant $\mathcal{S}_+^{M-1} = \{\boldsymbol{\lambda} \in \mathbb{R}^M_{\geq 0} : \sum_{m=1}^M (\lambda^{(m)})^2 = 1\}$. The inverse weight distribution $p(\boldsymbol{\lambda})$ controls the radial contribution for each point of A towards the HV. Existing work mainly considers a uniform distribution over the weights in order to compute the standard HV. We make the novel observation that we can assess the quality of the Pareto front in different regions of the objective space by varying this weight distribution. We call the resulting utility function the generalized hypervolume (GHV) indicator, denoted by U_{GHV}.

This GHV satisfies a weaker version of the Pareto complete property: $A \succeq B \implies U_{\text{GHV}}(A) \geq U_{\text{GHV}}(B)$. This property can be proved from the fact that the scalarization function $s_{\boldsymbol{\lambda}}(\mathbf{a})$ is a

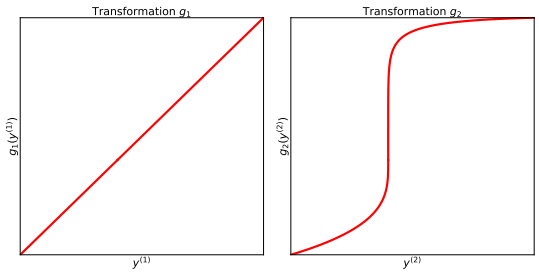
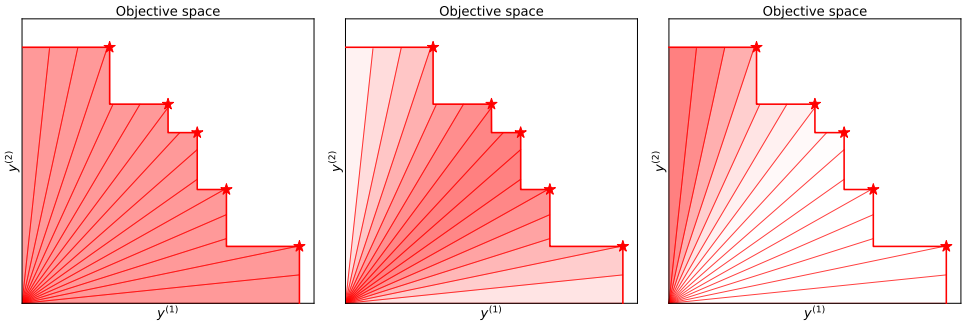


Figure 11: The transformation functions g_1 and g_2 used in Figure 4b.



(a) Beta$(1, 1)$ (b) Beta$(2.5, 2.5)$ (c) Beta$(5, 2)$

Figure 12: The contribution of each radial segment according to the distribution $p(\mathbf{w})$.

monotonic increasing function in $\mathbf{a} \in A$, which implies that the Pareto order is maintained. This performance criteria satisfies the strict Pareto complete property when all of the weights have non-zero density $p(\boldsymbol{\lambda}) > 0$. Intuitively, if a set of weights had zero density, then an improvement along these direction will not always result in a larger GHV. This means that we will not always be able to distinguish between a set which strictly dominates another, which is essential for strict Pareto completeness.

To generate different distributions $p(\boldsymbol{\lambda})$, we exploit the fact that the set $\mathcal{S}_+^{M-1}$ is isomorphic to the $(M-1)$-dimensional unit hypercube $[0,1]^{M-1}$. In particular, any point $\mathbf{w} \in [0,1]^{M-1}$ can be mapped onto the sphere to a point $\boldsymbol{\lambda}(\mathbf{w}) \in \mathcal{S}_+^{M-1}$ with

$$\begin{aligned}
\lambda^{(1)} &= \cos\left(\frac{\pi}{2}w_1\right) \\
\lambda^{(2)} &= \sin\left(\frac{\pi}{2}w_1\right)\cos\left(\frac{\pi}{2}w_2\right) \\
&\vdots \\
\lambda^{(M-1)} &= \sin\left(\frac{\pi}{2}w_1\right)\cdots\sin\left(\frac{\pi}{2}w_{M-2}\right)\cos\left(\frac{\pi}{2}w_{M-1}\right) \\
\lambda^{(M)} &= \sin\left(\frac{\pi}{2}w_1\right)\cdots\sin\left(\frac{\pi}{2}w_{M-2}\right)\sin\left(\frac{\pi}{2}w_{M-1}\right).
\end{aligned}$$

Using this result, we can use any distribution with a finite support to generate samples on $\mathcal{S}_+^{M-1}$. For example, we use $M-1$ independent Beta distribution Beta(a_m, b_m) to generate samples from $\mathbf{w} \in [0,1]^{M-1}$. In Figure 12, we present the radial contributions to the GHV for different weight distributions $p(\mathbf{w})$.

Remark. We are not the first to observe the importance of weighting the HV towards regions of interest. For example, an early paper [86] proposed another a weighted hypervolume indicator (WHV), $U_{\text{WHV}}(A) = \int_{\mathbb{R}^M} w(\mathbf{z})\mathbb{I}[\mathbf{r} \preceq \mathbf{z} \preceq A]d\mathbf{z}$, which introduces a weight function $w : \mathbb{R}^M \to \mathbb{R}$ directly into the integral. This approach is not as flexible as the one presented here because it relied on designing hand-crafted weight functions and an involved symbolic integration procedure.

I Experiments

I.1 Implementation details

All of the numerical results was implemented on Python 3.8 using open-source libraries: BoTorch (0.5.1) [3], GPyTorch (1.6.0) [26], NumPy (1.21.2), Pymoo (0.5.0) [8] [33], PyTorch (1.9.0) [56] and SciPy (1.7.3) [73]. All computations were performed on a computing cluster, where we restricted the computation to a single CPU core of an AMD EPYC 7742 64-Core Processor @ 2.25GHz. The code is available at `https://github.com/.../JES`.

Gaussian process model. For all the problems, we normalized the inputs and standardized the observations before performing Gaussian process regression. We assume an independent model

for each objective with a constant mean and Matérn 5/2 ARD kernel. Additionally, we assume a zero-mean Gaussian observation noise with a homogeneous variance $\boldsymbol{\sigma}(\mathbf{x}) = \boldsymbol{\sigma} \in \mathbb{R}^M$. For convenience, we rely on the default model in BoTorch, which additionally places Gamma priors for the kernel hyperparameters and observation variance. At each iteration of BO, we update the point estimates for the model hyperparameters and observation variance by maximizing the exact marginal log-likelihood using SciPy's default gradient optimizer.

Optimizing the acquisition function. To determine the next point, we optimize the acquisition function using multi-start L-BFGS-B [73]. We use the exact gradient inferred using automatic differentiation [56] for all algorithms except for the PES. The automatic gradients inferred for the PES acquisition function occasionally failed due to the differentiability issues arising from the damping procedure within the expectation propagation update [27]. As a result, we used the default 2-point finite difference method in SciPy to approximate the gradients of the PES acquisition function. To initialize the multi-start gradient optimizer, we evaluated the acquisition function on a space-filling design of $1000D$ random points. The starting points were then chosen as the best performing points from the initial check. The number of starting points that we used differed for each algorithm based on the overall expenses. For the sequential experiments, we used 20 starting points for PES, $5D$ starting points for the MES and JES, and $10D$ starting points for the rest. For the batch experiments, we used 20 starting points for PES and $5D$ starting points for the rest. For the information-theoretic acquisition functions, we could have initialized the starts near the sampled Pareto optimal points in order to promote faster convergence. We abstained from doing this because we wanted to give a fair comparison between the existing approaches. Specifically, we wanted to see how well information-based acquisition function worked out of the box without any optimization compared to the existing optimized approaches.

Sampling the Pareto points. To sample the Pareto optimal inputs and outputs, we first sample Gaussian process paths. We achieved this by using an approximate sampling strategy based on random Fourier feaures (Appendix A.3). We used the implementation in BoTorch and set the number of features to $L = 500$. To solve for the Pareto set and front, we used the multi-objective solver NSGA2 [18], which was implemented in Pymoo. We set the population size to be $N_{\text{pop}} = 100$, the number of generations to be $N_{\text{gen}} = 500$ and the number of offspring to be $N_{\text{off}} = 10$. In general, the multi-objective solver outputs an approximate set of Pareto optimal points with a size less than or equal to N_{pop}. To truncate this set into a size p, we used a HV truncation strategy. In particular, we greedily selected points based on their contribution to the sample HV generated by the sample $f_s(X_n)$—the reference point is set to the current estimate of the nadir minus some error $\hat{\mathbf{r}}_n - 0.1|\hat{\mathbf{r}}_n|$, where $\hat{r}_n^{(m)} = \min_{t=1,\dots,n} y_t^{(m)}$. This refinement strategy can be implemented quickly using the expected HV improvement strategy discussed in [14]. We implemented this sampling and optimizing procedure in sequence, but naturally this can be executed in parallel because we are performing S independent computations. As motivated by the wall times in Appendix I.6, implementing this step in parallel could be very beneficial computationally.

Box decompositions. We used the BoTorch implementation of the box decomposition strategy discussed in Algorithm 1 of [46]. The box decompositions were performed in sequence instead of being executed in parallel. From the wall times presented in Appendix I.6, the time required to compute the box decompositions becomes more demanding as the number of objectives increases.

Conditioning. We used the fantasizing feature in BoTorch to condition the current posterior on a collection of S independent Pareto samples of size p. We treated the Pareto samples as noisy pseudo-observations in the conditioning.

Acquisition functions. The benchmark comparison considers a range of popular acquisition functions that have all been implemented in BoTorch. We implemented our own version of the multi-objective PES, MES and JES. We now elaborate on the implementation details for each acquisition functions in the benchmark experiments.

- **TSEMO.** The Thomson sampling algorithm (TSEMO) [10] is a random acquisition function, which selects the point that maximizes the HV improvement according to a single sample of the Pareto front. Unlike the original paper [10], we select the point that improves the HV of the sample frontier $f_s(X_n)$ as opposed to the observation frontier generated by Y_n. We find this adjustment to

be sensible when there is observation noise. We modified the greedy batch extension in the same way.

- **ParEGO/ NParEGO.** The random scalarization strategy (ParEGO) [43] and its noisy counterpart (NParEGO) [15] are one of the most popular strategies for multi-objective BO. At each iteration, a random scalarization of the objectives is obtained by randomly drawing a weight $\mathbf{w} \in \mathbb{R}^M$ from the $(M-1)$-simplex. To target all the Pareto optimal points, we use the augmented Chebyshev scalarization: $\min_{m=1,\dots,M} w^{(m)} \tilde{f}^{(m)} + 0.01 \sum_{m=1} w^{(m)} \tilde{f}^{(m)}$. Here we have denoted $\tilde{f} \propto f$ as the objective function, which has been approximately normalized to $[0,1]^M$ using the observations Y_n. For the single objective problem, a Monte Carlo estimate of the expected improvement and the noisy expected improvement for the ParEGO and NParEGO algorithm are used respectively. The number of base samples for the Monte Carlo estimates is set to 128. For the batch setting, we sampled q different weights and optimized the acquisition function sequentially.

- **EHVI/ NEHVI.** The expected hypervolume improvement (EHVI) [14] and its noisy counterpart (NEHVI) [15] are an improvement-based acquisition function for the HV indicator. The number of base samples for the Monte Carlo estimates is set to 128. We set the reference point of the HV indicator to be equal to the observed nadir minus some error, $\hat{\mathbf{r}}_n - 0.1|\hat{\mathbf{r}}_n|$, where $\hat{r}_n^{(m)} = \min_{t=1,\dots,n} y_t^{(m)}$. This dynamic reference point strategy was recommended in the supplementary material of [14]. For the batch setting, we considered a sequentially greedy optimization strategy [14, 15].

- **PES.** The predictive entropy search (PES) [27, 29, 35, 36] acquisition function is approximated using expectation propagation. We implemented this algorithm from scratch in BoTorch under the guidance of the supplementary material in [27]. We used $S = 10$ Monte Carlo samples and $p = 10$ number of Pareto optimal inputs. We set the jitter for the matrix inversion to be 0.001 and the convergence threshold for the initialization stage to be 5% relative change in the mean and covariance. If the expectation propagation failed to converge, we outputted a random vector from the already sampled Pareto sets as the official recommendation. For the batch setting, we extended the approach outlined in [27]. This extension appears to be equivalent to the approach described in [29]. We optimized the resulting acquisition function using a joint optimization approach.

- **MES.** The maximum value entropy search (MES) [68, 74] can be approximated using all the conditional entropy estimates we devised in this paper. We used $S = 10$ Monte Carlo samples and $p = 10$ number of Pareto optimal outputs. For the Monte Carlo estimate MES-MC we set the number of base samples to 128. For the batch setting, we considered a greedy optimization strategy [14, 15]. For the batch setting, we consider the lower bound described by (12). We optimized the resulting acquisition function using a greedy optimization approach.

- **JES.** The joint entropy search (JES) [68, 74] can be approximated using all the conditional entropy estimates we devised in this paper. We used $S = 10$ Monte Carlo samples and $p = 10$ number of Pareto optimal points. For the Monte Carlo estimate JES-MC we set the number of base samples to 128. For the batch setting, we consider the lower bound described by (12). We optimized the resulting acquisition function using a greedy optimization approach.

Optimizing for the recommendation. To obtain the recommendation set, $\hat{\mathbb{X}}_n^*$, we used the NSGA2 multi-objective solver to optimize the posterior mean $\boldsymbol{\mu}_n$. Using the Pymoo implementation, we set the population size to be $N_{\text{pop}} = 500$, the number of generations to be $N_{\text{gen}} = 500$ and the number of offspring to be $N_{\text{off}} = 10$. We select the $p = 50$ points that greedily maximizes the HV generated by the mean objectives. The reference point for the HV truncation was set to the current estimate of the nadir minus some error: $\hat{\mathbf{r}}_n - 0.1|\hat{\mathbf{r}}_n|$, where $\hat{r}_n^{(m)} = \min_{t=1,\dots,n} y_t^{(m)}$. Before applying the truncation, we first augmented the Pareto set found at this iteration with the previous recommendation set $\hat{\mathbb{X}}_{n-1}^*$ in order to guarantee that some promising solutions weren't missed due to randomness of the solver.

Optimizing for the Pareto set. To obtain the baseline Pareto set, $\mathbb{X}^*$, we used the Pymoo's NSGA2 with a population size to be $N_{\text{pop}} = 1000$, the number of generations to be $N_{\text{gen}} = 5000$ and the number of offspring to be $N_{\text{off}} = 10$.

I.2 Benchmark problems

We initialized each test problem with with $2(D+1)$ training points using a random space filling design. The details of the individual benchmark problems is presented below.

ZDT2 (D=2, M=2, Noise=10%). The ZDT2 benchmark [85] is a bi-objective minimization problem $\min_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x})$, over the D-dimensional hypercube $\mathbb{X} = [0,1]^D$, where

$$f^{(1)}(\mathbf{x}) = x^{(1)}$$
$$f^{(2)}(\mathbf{x}) = g(\mathbf{x})\left(1 - \left(\frac{f^{(1)}(\mathbf{x})}{g(\mathbf{x})}\right)^2\right)$$

with $g(\mathbf{x}) = 1 - \frac{9}{D-1}\sum_{i=2}^{D} x^{(i)}$. For the experiments, we considered the maximization problem by negating the objective: $\max_{\mathbf{x}\in\mathbb{X}}(-f(\mathbf{x}))$. The standard deviation of the Gaussian observation noise is set to $\boldsymbol{\sigma}^{1/2} = (0.1, 0.8)$, which is estimated to be around 10% the range of the objectives from 1000 function evaluations. For the HV indicators we use the reference point of $\mathbf{r} = (-11, -11)$.

SnAr (D=4, M=2, Noise=3%). The SnAr benchmark [23] is a bi-objective minimization problem $\min_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x})$, which is defined over a four-dimensional space, $\mathbb{X} = [0.5, 2.0]\times[1.0, 5.0]\times[0.1, 0.5]\times[30, 120]$, where

$$f^{(1)}(\mathbf{x}) = -\log(\texttt{STY}(\mathbf{x}))$$
$$f^{(2)}(\mathbf{x}) = \log(\texttt{E-Factor}(\mathbf{x}))$$

with `STY` and `E-Factor` denoting the space-time yield and environmental factor, respectively. The functions `STY` and `E-Factor` depend on the solution of an ordinary differential equation governed by a kinetic model, where the rate constants have been estimated by empirical tests [41]. Unlike the original paper [23], we have taken the logarithm of the output to accommodate for the additive noise in the objective space. For the experiments, we considered the maximization problem by negating the objective: $\max_{\mathbf{x}\in\mathbb{X}}(-f(\mathbf{x}))$. The standard deviation of the Gaussian observation noise is set to $\boldsymbol{\sigma}^{1/2} = (0.12, 0.08)$, which is estimated to be around 3% the range of the objectives from 1000 function evaluations. For the HV indicators we use the reference point of $\mathbf{r} = (5.5, -5)$.

Penicillin (D=7, M=3, Noise=1%). The Penicillin benchmark [47] is a three objective minimization problem $\min_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x})$, which is defined over a seven-dimensional space, $\mathbb{X} = [60, 120] \times [0.05, 18] \times [293, 303] \times [0.05, 18] \times [0.01, 0.5] \times [500, 700] \times [5, 6.5]$, where

$$f^{(1)}(\mathbf{x}) = -\texttt{PenicillinConcentration}(\mathbf{x})$$
$$f^{(2)}(\mathbf{x}) = \texttt{CO2Concentration}(\mathbf{x}))$$
$$f^{(3)}(\mathbf{x}) = \texttt{TimeToFerment}(\mathbf{x})$$

with `PenicillinConcentration`, `CO2Concentration` and `TimeToFerment` denoting the concentration of the desirable product, the concentration of the subproduct and the time to ferment, respectively. The equations describing the pharmaceutical simulation are given in the original reference [47]. We used the open-source implementation available in BoTorch [3, 17]. For the experiments, we considered the maximization problem by negating the objective: $\max_{\mathbf{x}\in\mathbb{X}}(-f(\mathbf{x}))$. The standard deviation of the Gaussian observation noise is set to $\boldsymbol{\sigma}^{1/2} = (0.14, 0.8, 3.8)$, which is estimated to be around 1% the range of the objectives from 1000 function evaluations. For the HV indicators we use the reference point of $\mathbf{r} = (-1.85, -86.93, -514.70)$.

Marine Design (D=6, M=4, Noise=0.5%). This marine design benchmark [55, 61, 71] is a four objective minimization problem $\min_{\mathbf{x}\in\mathbb{X}} f(\mathbf{x})$, which is defined over a six-dimensional space, $\mathbb{X} = [150, 274.32] \times [20, 32.31] \times [13, 25] \times [10, 11.71] \times [14, 18] \times [0.63, 0.75]$, where

$$f^{(1)}(\mathbf{x}) = \texttt{TransportationCost}(\mathbf{x})$$
$$f^{(2)}(\mathbf{x}) = \texttt{Weight}(\mathbf{x})$$
$$f^{(3)}(\mathbf{x}) = -\texttt{AnnualCargo}(\mathbf{x})$$

$$f^{(3)}(\mathbf{x}) = \texttt{SumOfConstraints}(\mathbf{x})$$

with `TransportationCost`, `Weight`, `AnnualCargo` and `SumOfConstraints` denoting the transportation cost, the ship weight, the annual cargo transport capacity and the sum of the constraint violations, respectively. The formal equations describing the functions are presented in the original references [55, 61, 71]. For the experiments, we considered the maximization problem by negating the objective: $\max_{\mathbf{x}\in\mathbb{X}}(-f(\mathbf{x}))$. The standard deviation of the Gaussian observation noise is set to $\boldsymbol{\sigma}^{1/2} = (10, 77, 132, 0.07)$, which is estimated to be around 0.5% the range of the objectives from 1000 function evaluations. For the HV indicators we use the reference point of $\mathbf{r} = (250, -20000, -25000, -15.0)$.

I.3 Sensitivity analysis

In the section we empirically analyse how sensitive the approximations to the information-theoretic acquisition functions are with different choices of Monte Carlo samples S and number of Pareto optimal points p. For the experiments in the main section, we set $S = 10$ and $p = 10$ for all information-theoretic acquisition functions. This selection is comparable to the existing literature [4, 27, 68, 74]. For the sake of brevity we present results only for one benchmark problem: ZDT2 (D=2, M=2, Noise=10%, q=1).

To test the sensitivity with regards to the number of Monte Carlo samples S, we fix $p = 10$ and ran the benchmark problem 100 times with $S \in \{1, 5, 10, 25, 50\}$. To test the sensitivity with regards to the number of Pareto optimal points p, we fix $S = 10$ and ran the benchmark problem 100 times with $p \in \{1, 5, 10, 25, 50\}$. We report the median log HV discrepancy over the runs in Figure 13 and Figure 14. We report the wall times in Figure 15. For the log HV plots, we did not include error bars because it is very hard to visually discriminate between the results. We analysed the upper and lower quartile plots and they showed a similar ranking to the plots displayed below. Some experiments could not be performed because of the unwieldy space and time requirements.

Performance-wise there does not appear to be much variation for acquisition function when $S > 1$ and $p > 1$. Naturally the wall times increase with the number of Monte Carlo samples S because the sampling and optimization of the Gaussian process paths are done in sequence (this step could be parallelized in practice). Overall, for this example problem, there does not appear to be much benefit in using a larger number of Monte Carlo samples S or Pareto optimal points p.

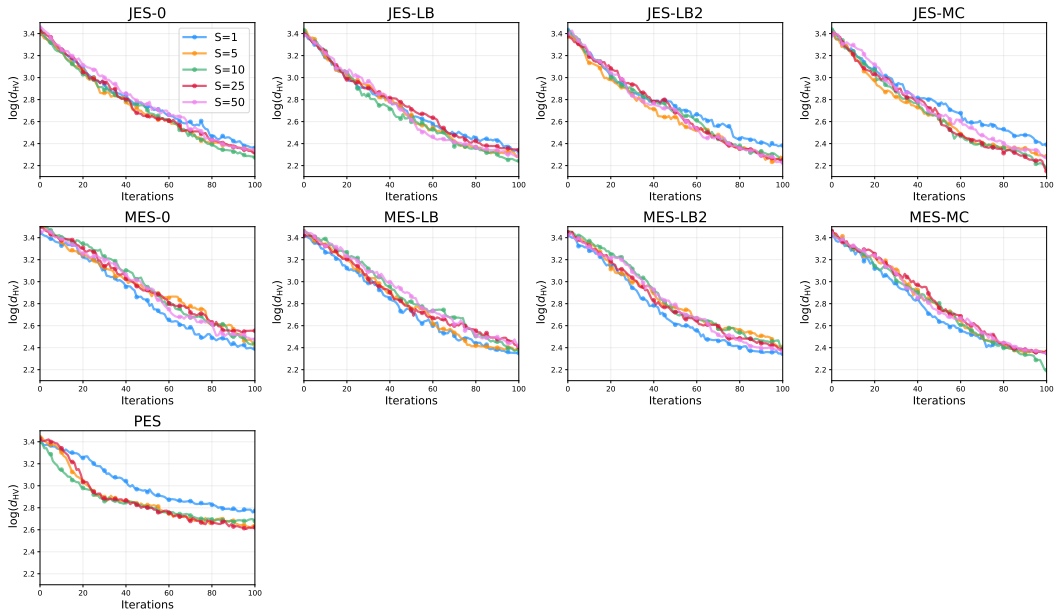


Figure 13: A comparison of the median logarithm HV discrepancy over different number of Monte Carlo samples S.

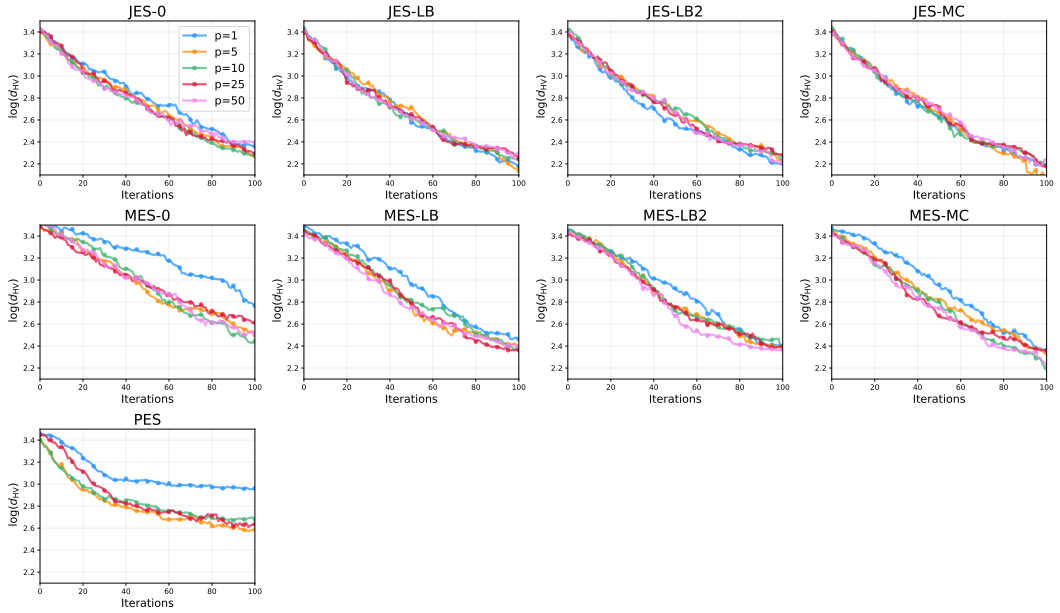


Figure 14: A comparison of the median logarithm HV discrepancy over different number of Pareto optimal samples p.

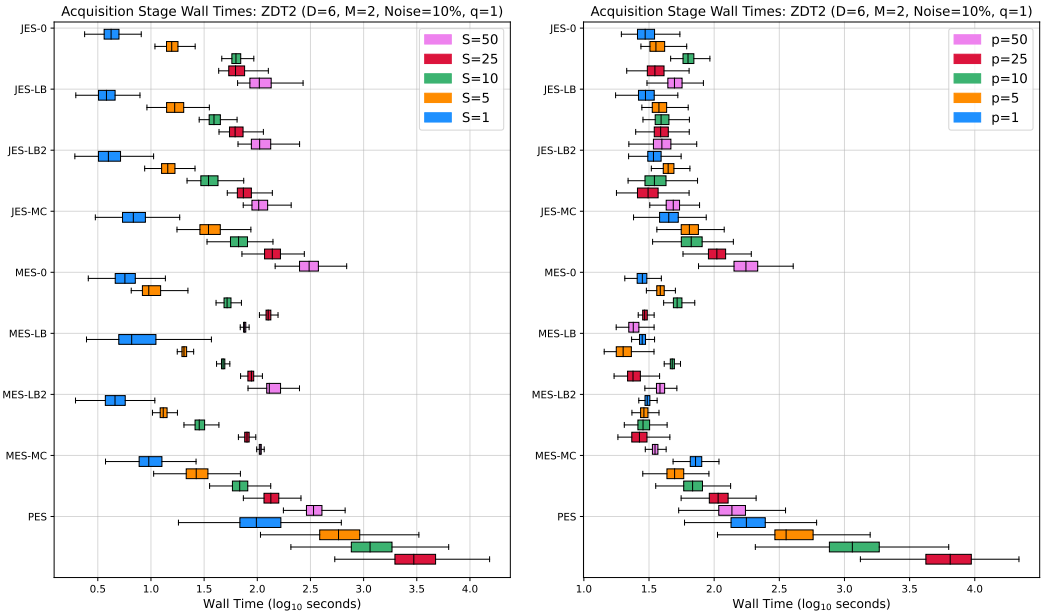


Figure 15: A box-plot comparison of the wall times for the acquisition stage over different number of Monte Carlo samples (left) and the number of Pareto optimal samples (right). The box-plots include the median, interquartile range and the extreme values after excluding the outliers. The acquisition stage includes any initialization computations such as the box decompositions and sampling the Pareto optimal points—it does not include initializing the posterior model. All of the runs of each algorithm was performed on a computing cluster, where we restricted the computation to a single CPU core of an AMD EPYC 7742 64-Core Processor @ 2.25GHz.

I.4 Sequential experiments (q=1)

For the sequential experiments, we have produced many different plots to compare the performance of the various algorithm. For all the experiments, we have plotted the log HV discrepancy at the mean and the quartiles. We also include the log GHV discrepancy plots, which considers the HV under different weight distributions as described in Appendix H. We also include plots of the Pareto front approximation in the objective space.

I.4.1 ZDT2 (D=2, M=2, Noise=10%, q=1)

From the plots below, we see that all the BO algorithms are all capable of targetting the points at top right of the Pareto front. Notably, these points contributes the most towards the standard HV indicator. The Thompson sampling and information-theoretic driven algorithms are able to a more diverse range of points on the Pareto front for this test problem.

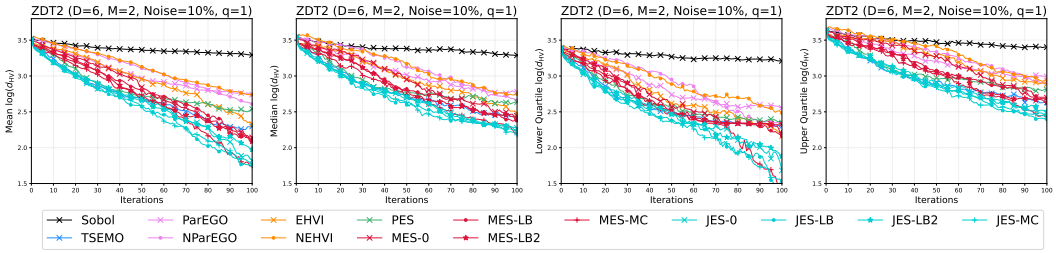


Figure 16: From left to right, we present a comparison of the mean, median, lower quartile and upper quartile logarithm HV discrepancy, respectively.

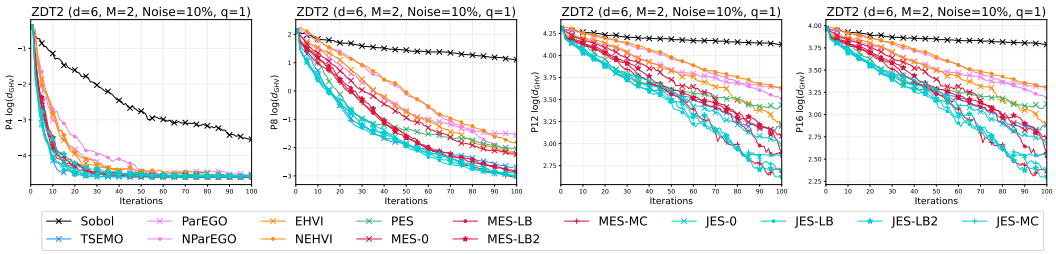


Figure 17: From left to right, we present a comparison of the mean logarithm GHV discrepancy for the weight distributions P4, P8, P12 and P16 (described in Figure 18), respectively.

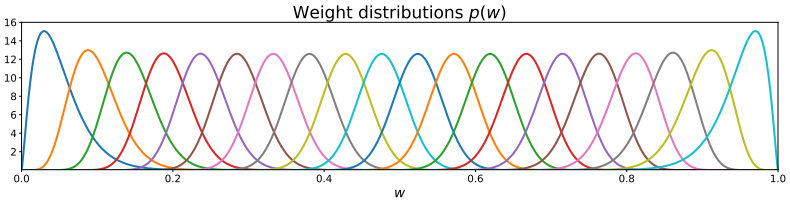


Figure 18: The Beta distributions used to generate the weights for the logarithm GHV discrepancy.

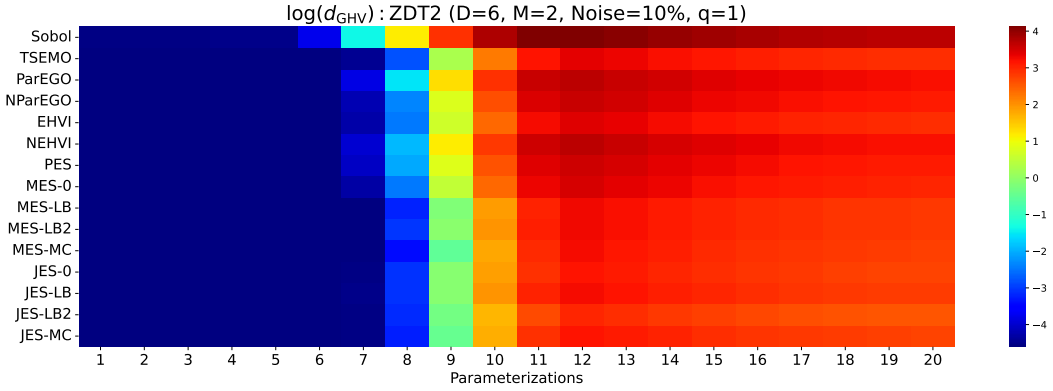


Figure 19: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 18.

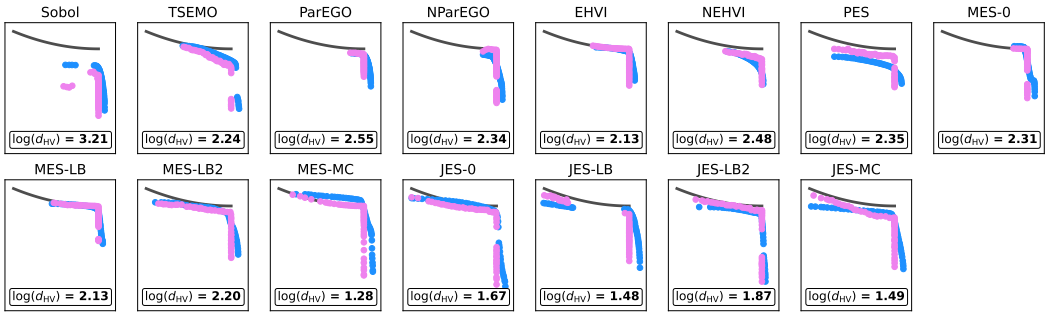


Figure 20: A comparison of the lower quartile Pareto fronts obtained at the final time $N = 100$. The true Pareto front $f(\mathbb{X}^*)$ is denoted by the black line, the Pareto front of the model posterior means at the recommended points $\boldsymbol{\mu}_N(\hat{\mathbb{X}}^*)$ are coloured blue and the actual objectives values at recommended points $f(\hat{\mathbb{X}}^*)$ are coloured pink.

I.4.2 SnAr (D=4, M=2, Noise=3%, q=1)

Almost all of the BO algorithms performed similarly on this benchmark—the only weak performer was PES. Despite achieving small HV discrepancy, Figure 24 illustrates that many of the BO approaches struggle to identify the shape of the true Pareto front. When looking at the various parameterization in Figure 23, we see that the lower bound estimates MES-LB and JES-LB performed slightly better than the rest, especially in the left region of the Pareto front, where the second objective is preferred more than the first.

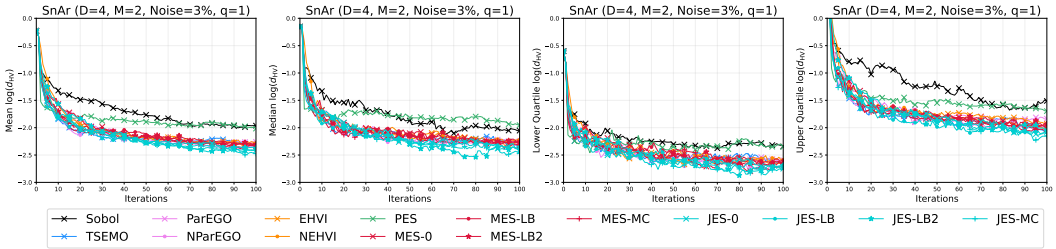


Figure 21: From left to right, we present a comparison of the mean, median, lower quartile and upper quartile logarithm HV discrepancy, respectively

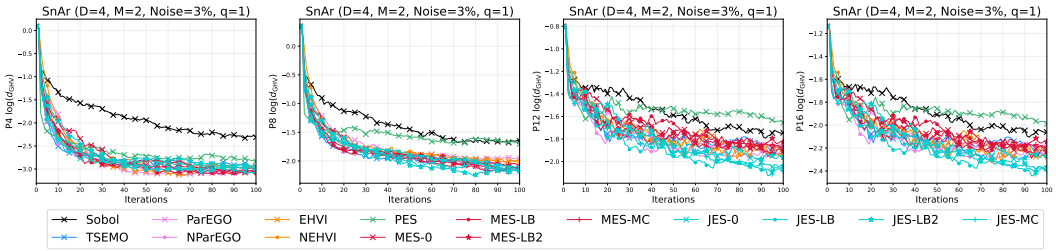


Figure 22: From left to right, we present a comparison of the mean logarithm GHV discrepancy for the weight distributions P4, P8, P12 and P16 (described in Figure 18), respectively.

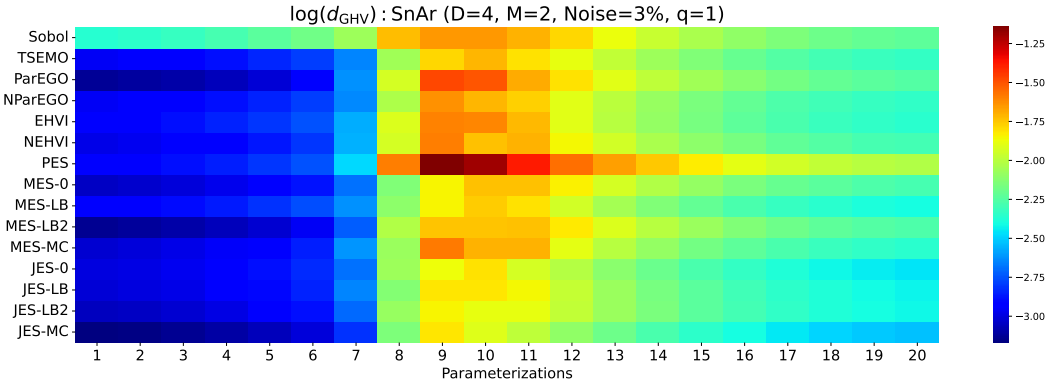


Figure 23: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 18.

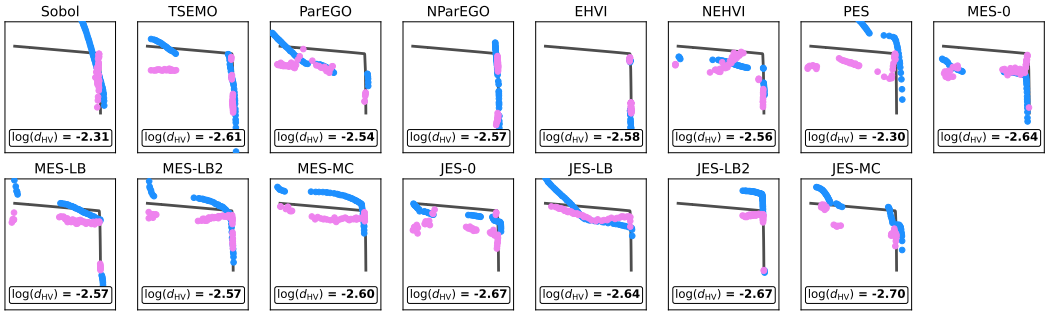


Figure 24: A comparison of the lower quartile Pareto fronts obtained at the final time $N = 100$. The true Pareto front $f(\mathbb{X}^*)$ is denoted by the black line, the Pareto front of the model posterior means at the recommended points $\boldsymbol{\mu}_N(\hat{\mathbb{X}}^*)$ are coloured blue and the actual objectives values at recommended points $f(\hat{\mathbb{X}}^*)$ are coloured pink.

I.4.3 Penicillin (D=7, M=3, Noise=1%, q=1)

The BO algorithms all performed similarly on this benchmark problems. The HV plots gives an indication to how difficult the underlying objective function is to model using a small number of observations. The best performers in terms of the final standard HV discrepancy were TSEMO, NParEGO, NEHVI and JES. From examining Figure 28, it is clear that the HV is dominated by the third objective, which has a considerably larger range than the other objectives. Figure 26 shows that PES actually outperforms the other algorithms in identifying many parts of the Pareto front that do not contribute that much to the standard HV calculation. This example again illustrates the difficulty in assessing multi-objective algorithms using only a single criterion.

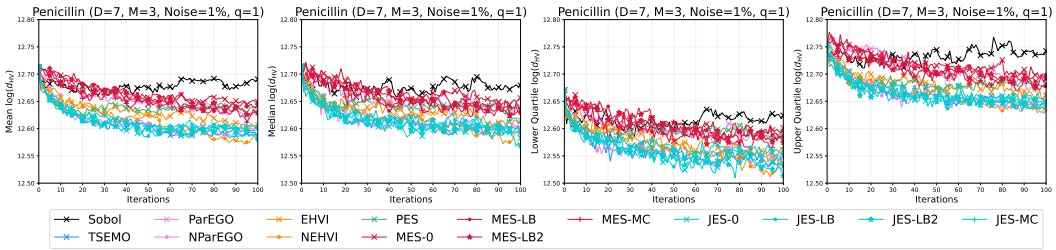


Figure 25: From left to right, we present a comparison of the mean, median, lower quartile and upper quartile logarithm HV discrepancy, respectively.

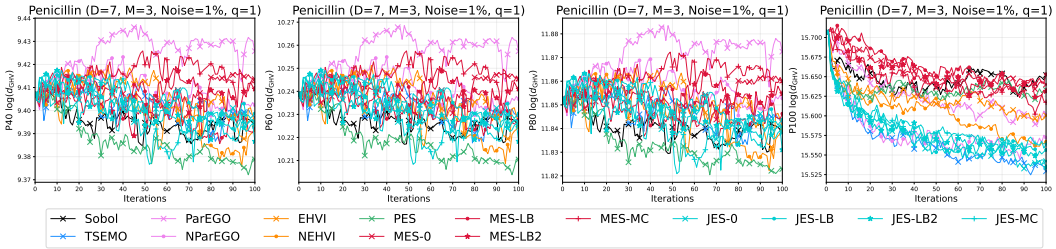


Figure 26: From left to right, we present a comparison of the mean logarithm GHV discrepancy for the weight distributions P40, P60, P80 and P100 (described in Figure 27), respectively.

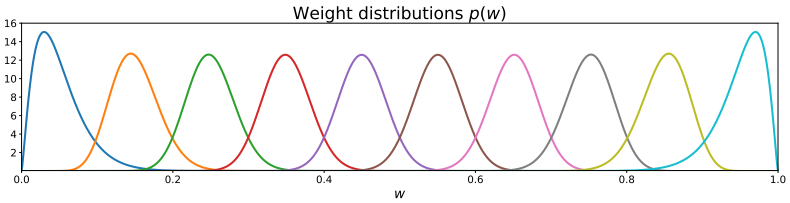


Figure 27: The Beta distributions used to generate the weights for the logarithm GHV discrepancy.

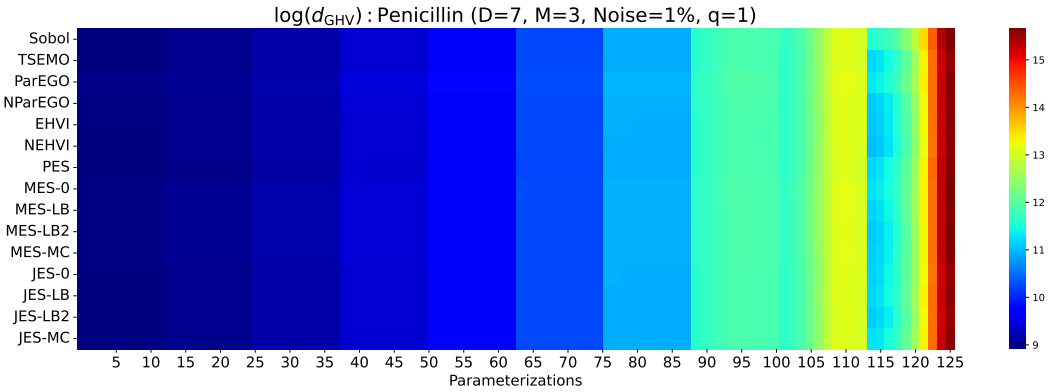


Figure 28: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 27.

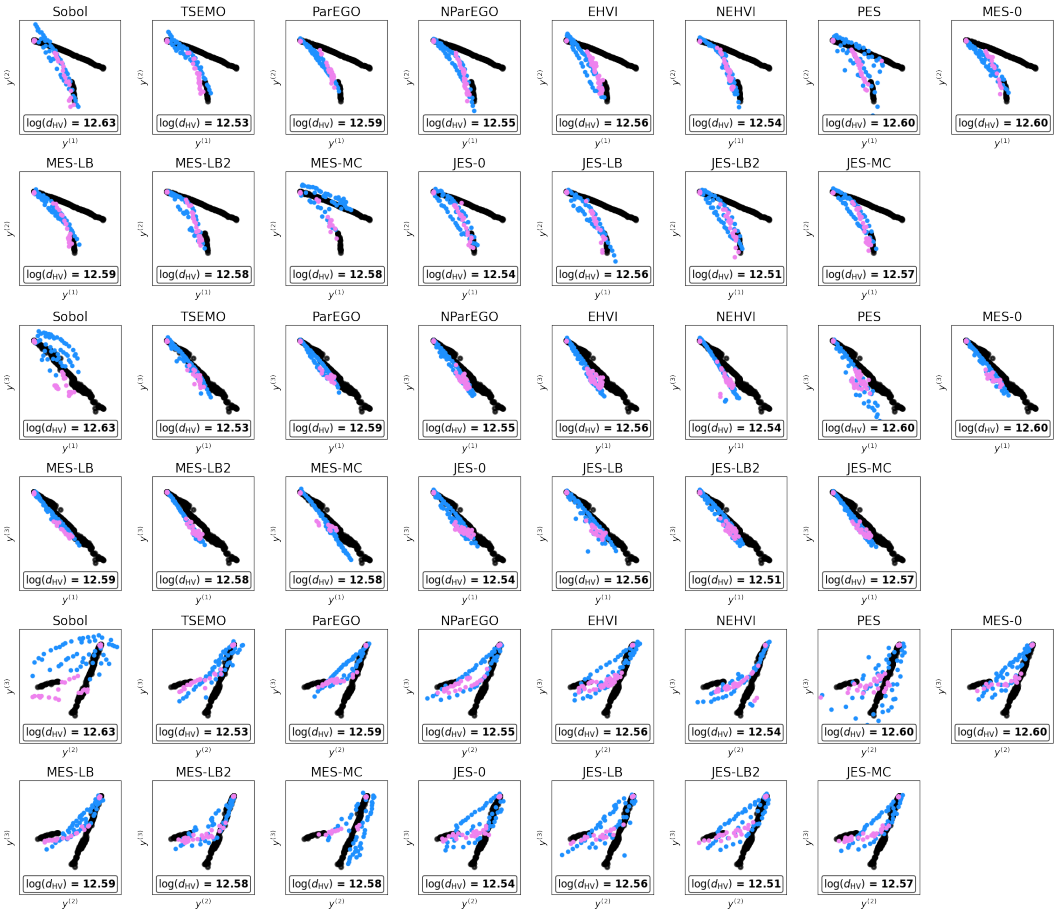


Figure 29: A comparison of the lower quartile Pareto fronts obtained at the final time $N = 100$. The points corresponding to the true Pareto front $f(\mathbb{X}^*)$ are coloured black, the Pareto front of the model posterior means at the recommended points $\boldsymbol{\mu}_N(\hat{\mathbb{X}}^*)$ are coloured blue and the actual objectives values at recommended points $f(\hat{\mathbb{X}}^*)$ are coloured pink.

I.4.4 Marine (D=6, M=4, Noise=0.5%, q=1)

For this benchmark, the lower bound JES approximation is the best in terms of the standard HV. The noiseless estimates MES-0 and JES-0 perform much worse than the other conditional entropy approximations. We omitted the experiments for the MES-MC and JES-MC algorithms because they were too expensive to compute. From viewing the Pareto front plots Figure 33, it is clear that this the noiseless approximations are more greedy than their counterparts.

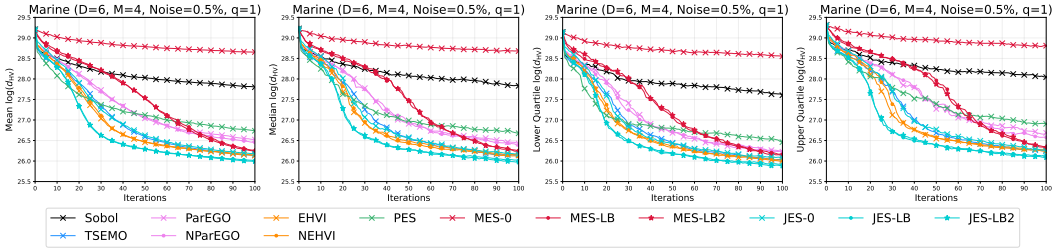


Figure 30: From left to right, we present a comparison of the mean, median, lower quartile and upper quartile logarithm HV discrepancy, respectively.

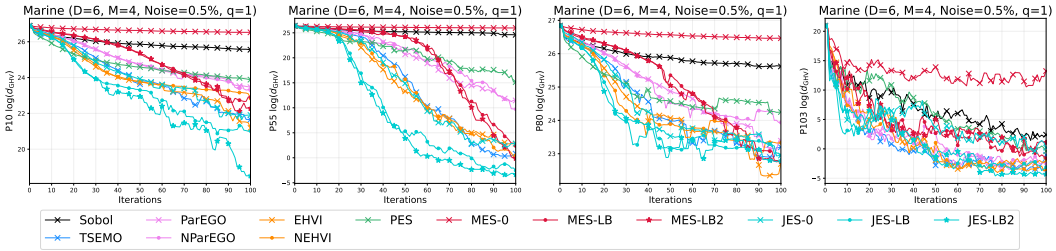


Figure 31: From left to right, we present a comparison of the mean logarithm GHV discrepancy for the weight distributions P10, P55, P80 and P103 (described in Figure 32), respectively.

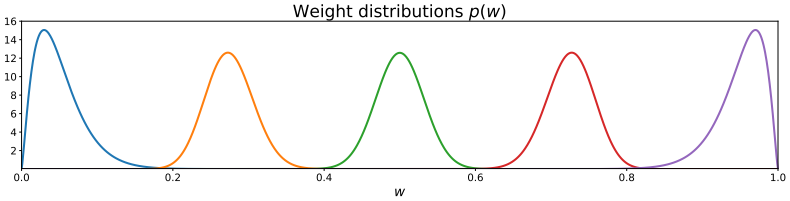


Figure 32: The Beta distributions used to generate the weights for the logarithm GHV discrepancy.

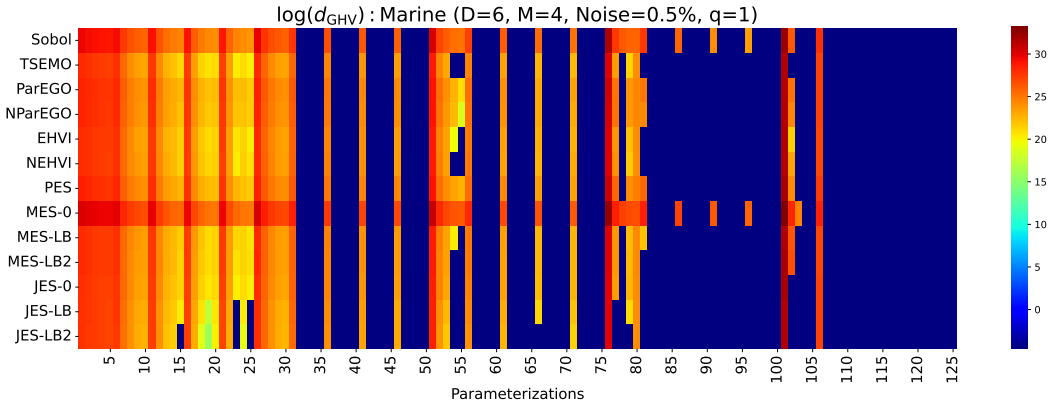


Figure 33: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 27.

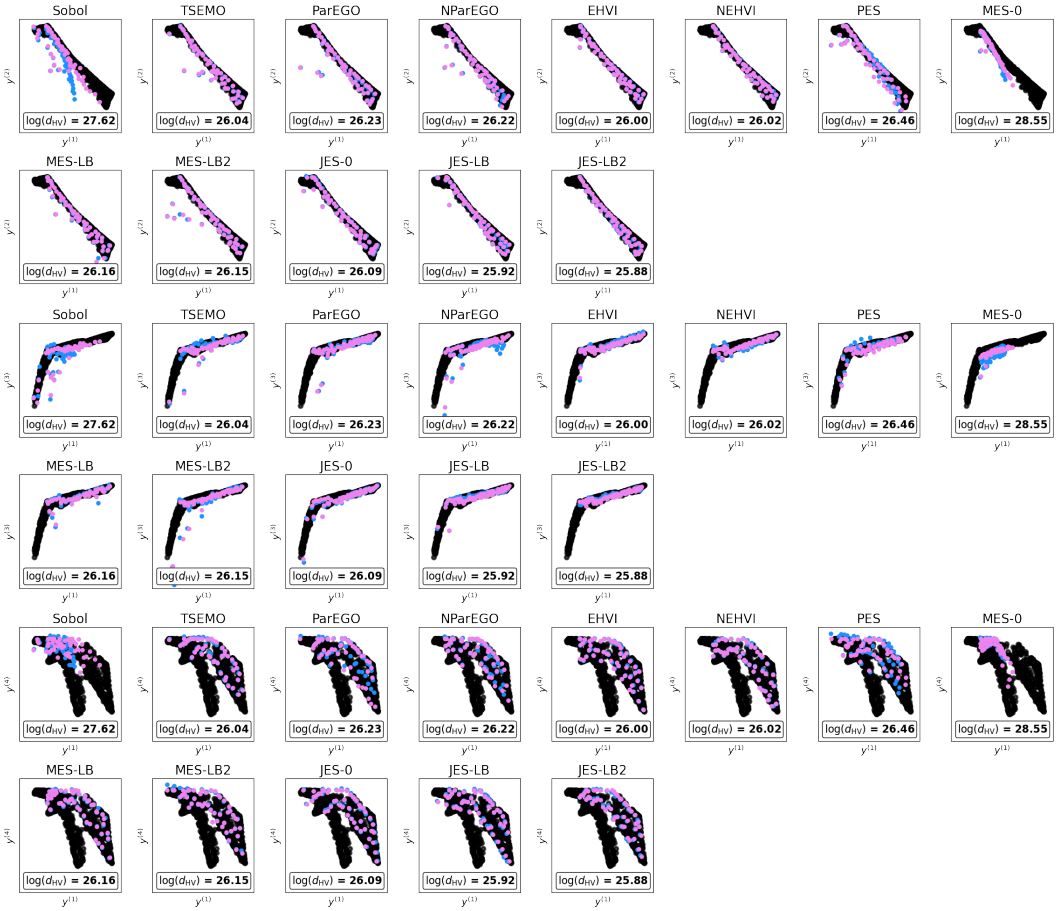


Figure 33: A comparison of the lower quartile Pareto fronts obtained at the final time $N = 100$. The points corresponding to the true Pareto front $f(\mathbb{X}^*)$ are coloured black, the Pareto front of the model posterior means at the recommended points $\boldsymbol{\mu}_N(\hat{\mathbb{X}}^*)$ are coloured blue and the actual objectives values at recommended points $f(\hat{\mathbb{X}}^*)$ are coloured pink.

I.5 Batch experiments (q=8)

In the previous section we considered the sequential setting where only $q = 1$ objective function evaluation was allowed to take place at each time. In this section, we consider the batch setting, where we allow $q = 8$ function evaluations to take place at each time.

I.5.1 ZDT2 (D=2, M=2, Noise=10%, q=8)

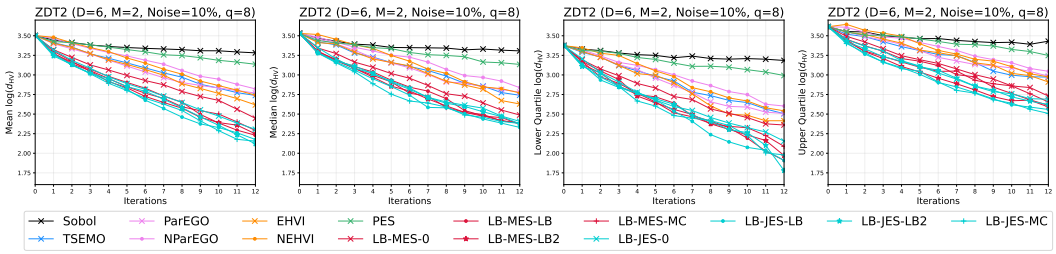


Figure 34: From left to right, we present a comparison of the mean, median, lower quartile and upper quartile logarithm HV discrepancy, respectively.

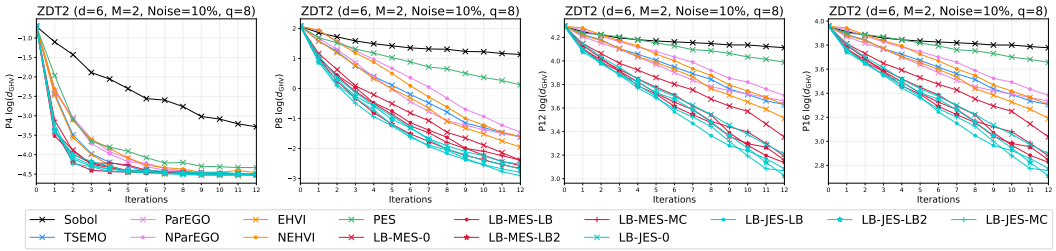


Figure 35: From left to right, we present a comparison of the mean logarithm GHV discrepancy for the weight distributions P4, P8, P12 and P16 (described in Figure 18), respectively.

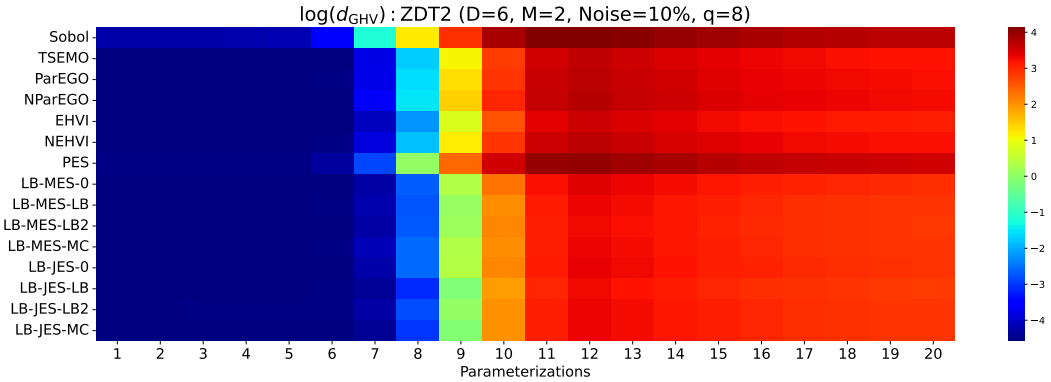


Figure 36: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 18.

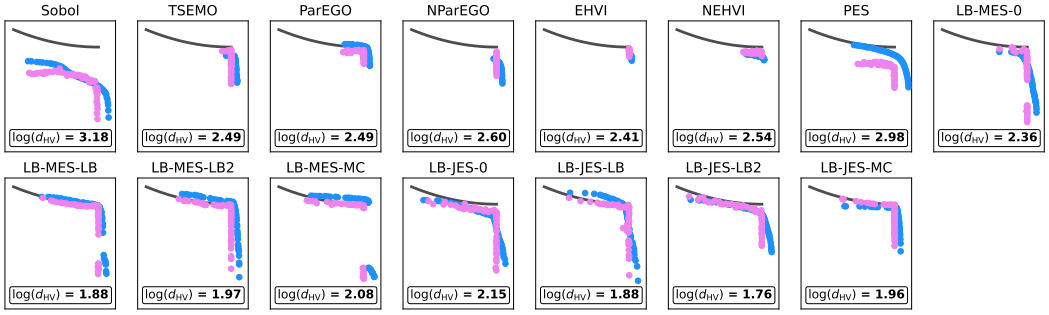


Figure 37: A comparison of the lower quartile Pareto fronts obtained at the final time $N = 100$. The true Pareto front $f(\mathbb{X}^*)$ is denoted by the black line, the Pareto front of the model posterior means at the recommended points $\boldsymbol{\mu}_N(\hat{\mathbb{X}}^*)$ are coloured blue and the actual objectives values at recommended points $f(\hat{\mathbb{X}}^*)$ are coloured pink.

I.5.2 SnAr (D=4, M=2, Noise=3%, q=8)

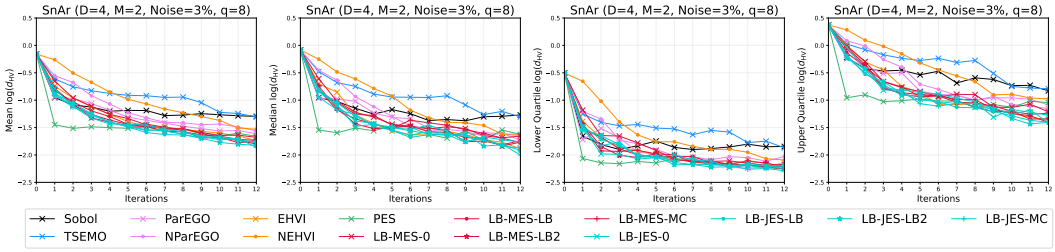


Figure 38: From left to right, we present a comparison of the mean, median, lower quartile and upper quartile logarithm HV discrepancy, respectively.

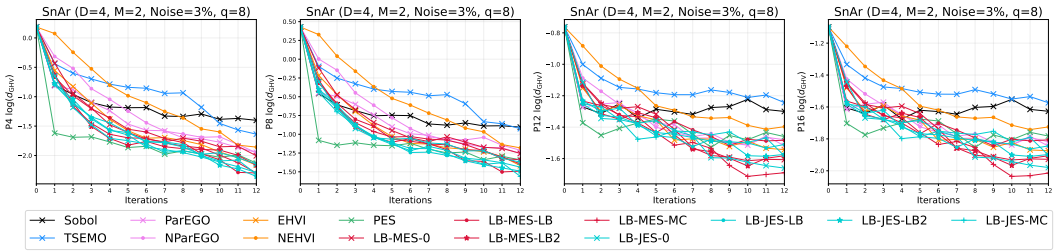


Figure 39: From left to right, we present a comparison of the mean logarithm GHV discrepancy for the weight distributions P4, P8, P12 and P16 (described in Figure 18), respectively.

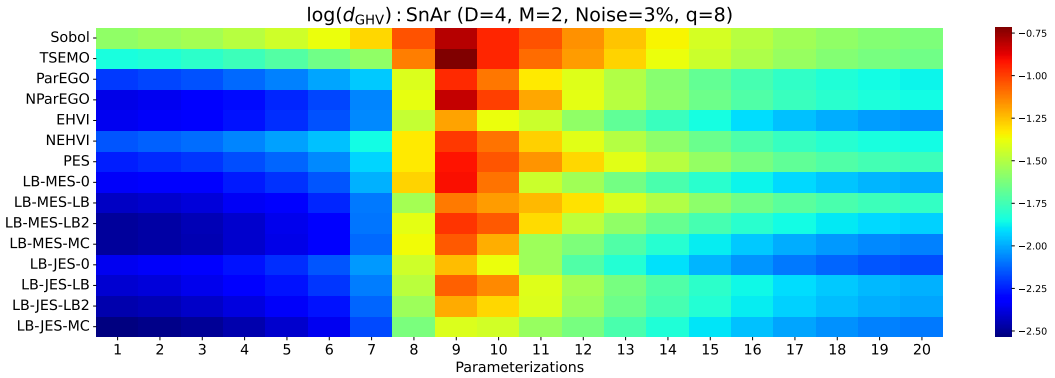


Figure 40: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 18.

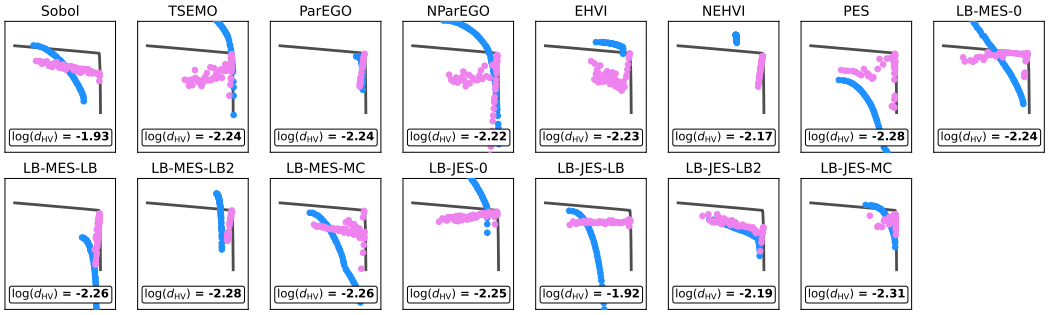


Figure 41: A comparison of the lower quartile Pareto fronts obtained at the final time $N = 100$. The true Pareto front $f(\mathbb{X}^*)$ is denoted by the black line, the Pareto front of the model posterior means at the recommended points $\boldsymbol{\mu}_N(\hat{\mathbb{X}}^*)$ are coloured blue and the actual objectives values at recommended points $f(\hat{\mathbb{X}}^*)$ are coloured pink.

I.5.3 Penicillin (D=7, M=3, Noise=1%, q=8)

We omitted the experiments for the qLB-MES-MC and qLB-JES-MC algorithms because they were too expensive to compute.

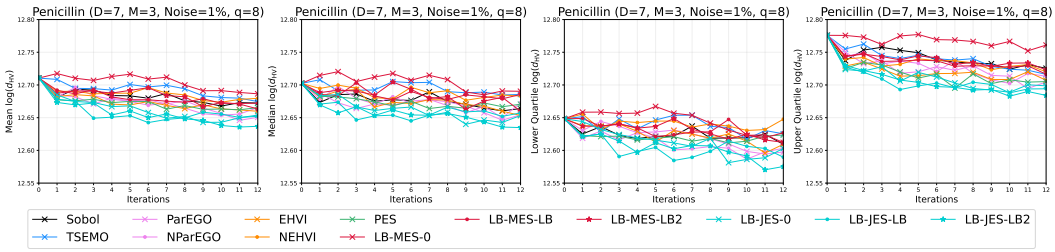


Figure 42: From left to right, we present a comparison of the mean, median, lower quartile and upper quartile logarithm HV discrepancy, respectively.

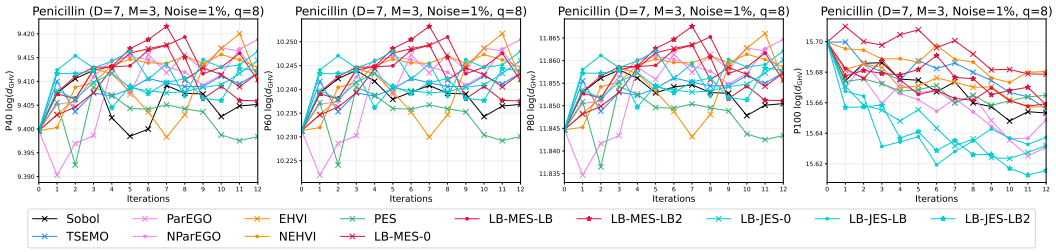


Figure 43: From left to right, we present a comparison of the mean logarithm GHV discrepancy. The weight distributions P40, P60, P80 and P100 (described in Figure 27), respectively.

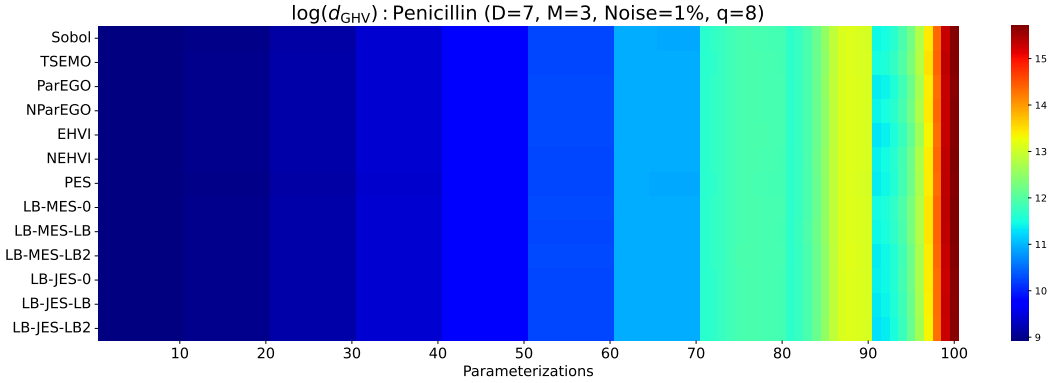
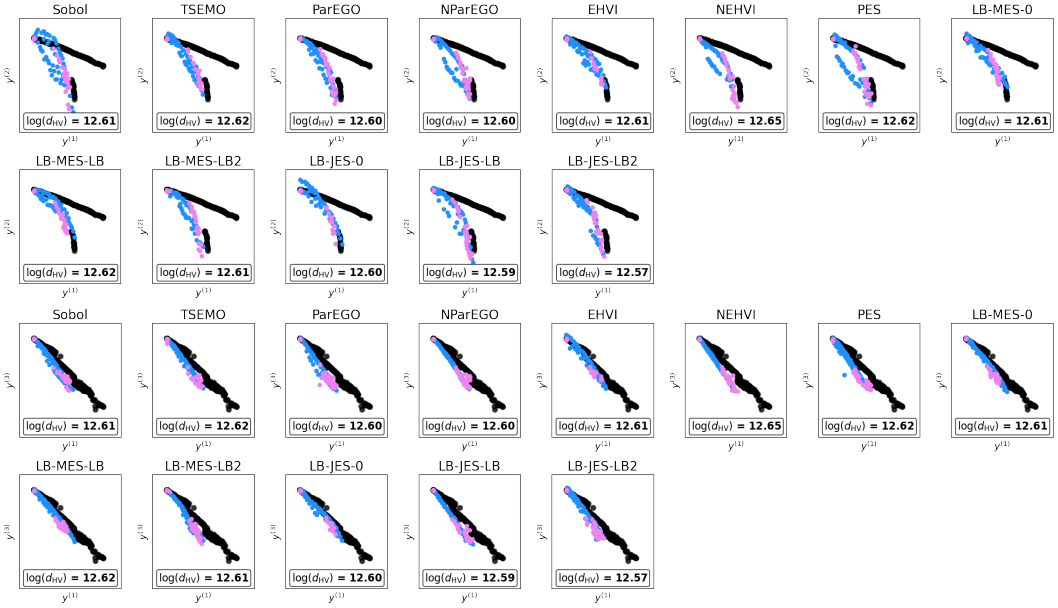


Figure 44: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 27.



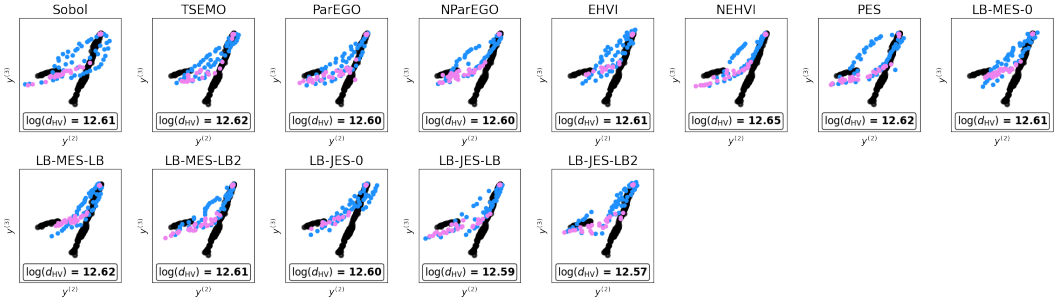


Figure 44: A comparison of the lower quartile Pareto fronts obtained at the final time $N = 100$. The points corresponding to the true Pareto front $f(\mathbb{X}^*)$ are coloured black, the Pareto front of the model posterior means at the recommended points $\boldsymbol{\mu}_N(\hat{\mathbb{X}}^*)$ are coloured blue and the actual objectives values at recommended points $f(\hat{\mathbb{X}}^*)$ are coloured pink.

I.5.4 Marine (D=6, M=4, Noise=0.5%, q=8)

We omitted the experiments for the qEVHI, qLB-MES-MC and qLB-JES-MC algorithms because they were too expensive to compute.

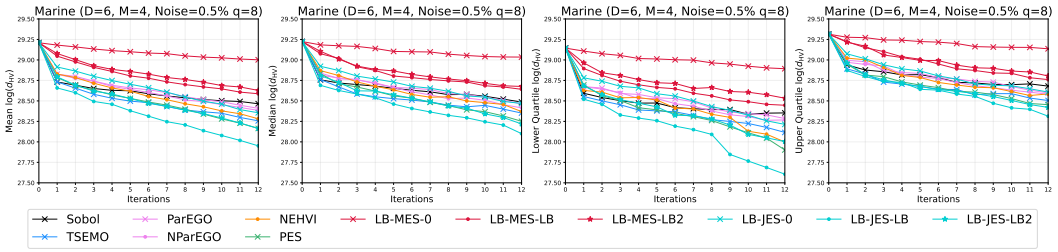


Figure 45: From left to right, we present a comparison of the mean, median, lower quartile and upper quartile logarithm HV discrepancy, respectively.

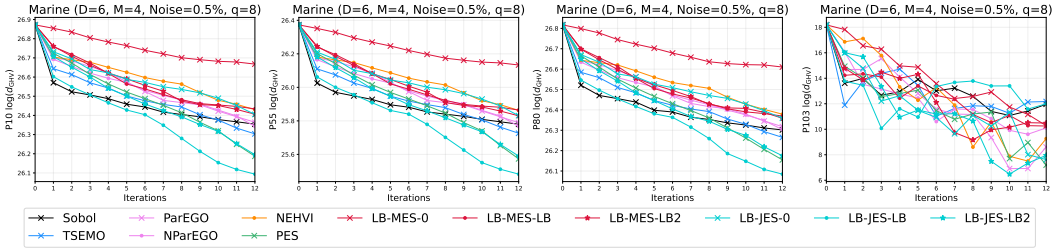


Figure 46: From left to right, we present a comparison of the mean logarithm GHV discrepancy for the weight distributions P10, P55, P80 and P103 (described in Figure 32), respectively.

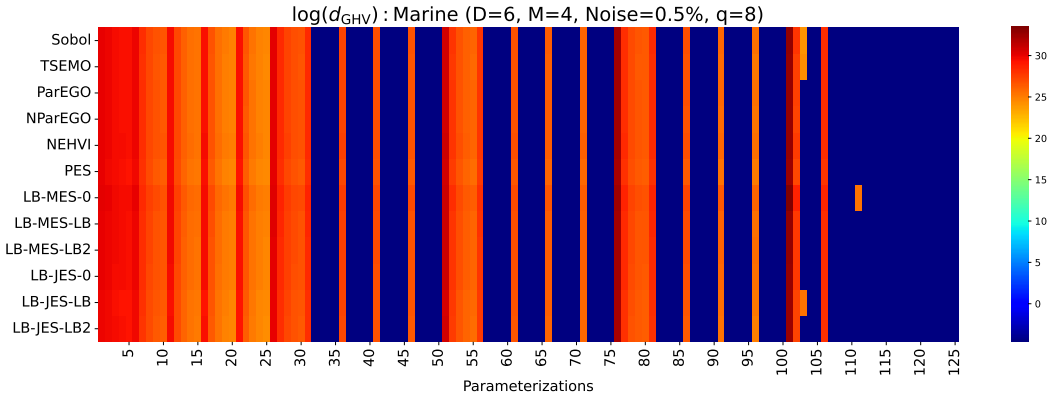


Figure 47: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 27.

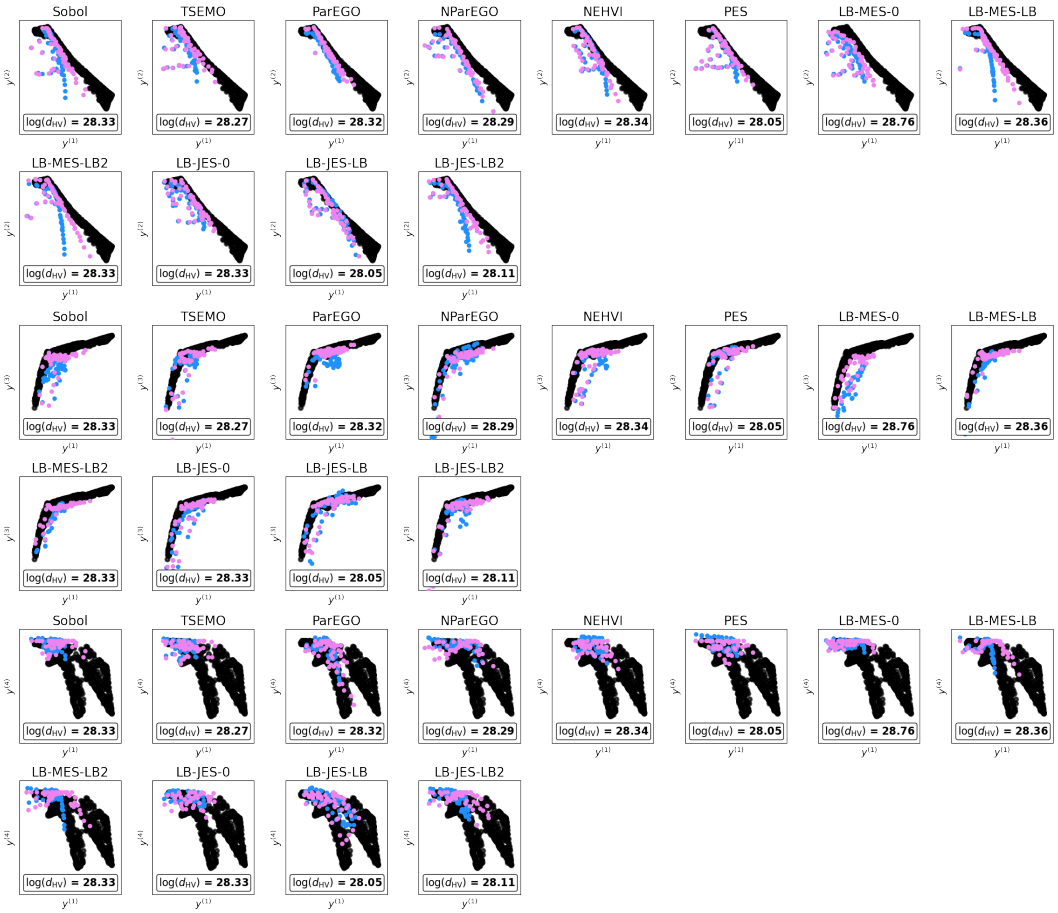


Figure 47: A comparison of the lower quartile Pareto fronts obtained at the final time $N = 100$. The points corresponding to the true Pareto front $f(\mathbb{X}^*)$ are coloured black, the Pareto front of the model posterior means at the recommended points $\boldsymbol{\mu}_N(\hat{\mathbb{X}}^*)$ are coloured blue and the actual objectives values at recommended points $f(\hat{\mathbb{X}}^*)$ are coloured pink.

I.6 Wall times

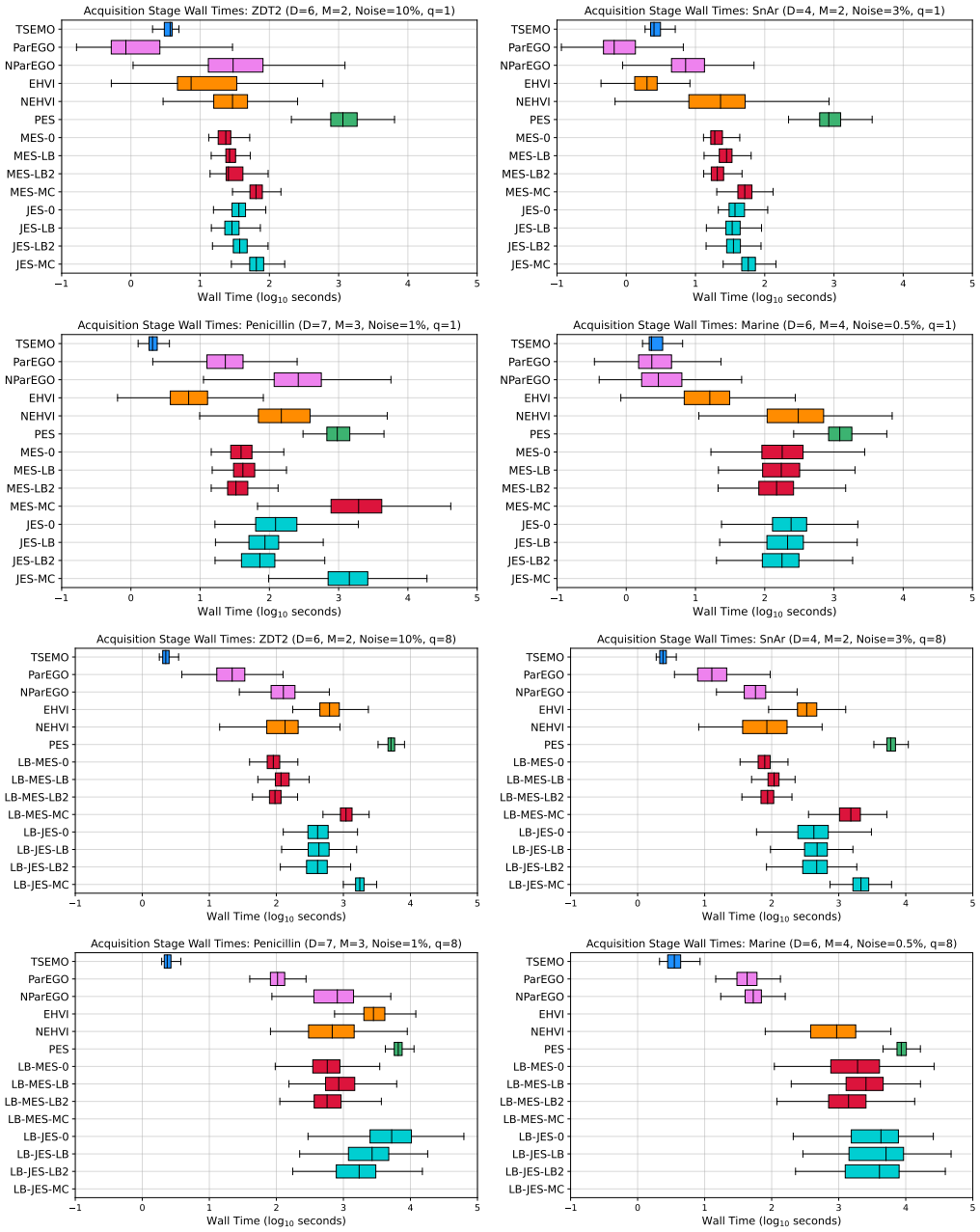


Figure 48: A box-plot comparison of the wall times for the acquisition stage, which includes the median, interquartile range and the extreme values after excluding the outliers. The acquisition stage includes any initialization computations such as the box decompositions and sampling the Pareto optimal points—it does not include initializing the posterior model. All of the runs of each algorithm was performed on a computing cluster, where we restricted the computation to a single CPU core of an AMD EPYC 7742 64-Core Processor @ 2.25GHz.

Figure 49: A bar chart comparison of the mean wall times proportions for the information-theoretic acquisition stage. The initialization step includes generation of base samples for the Monte Carlo estimates, the conditioning for JES and the initial expectation propagation steps for PES. The Pareto sampling and box decompositions steps are executed in sequence, hence the proportion of these parts could be reduced if these steps were executed in parallel. As the number of objectives increases, the one time box decomposition is the dominant contributor to the wall time for the MES and JES. The dominant cost of the PES is always the gradient optimization step because we estimate the gradients using finite differences, which is both expensive and inefficient.

J Extensions

The discussion so far focussed on the problem of unconstrained multi-objective optimization where evaluations of the vector-valued function are executed individually or in batch. Not all multi-objective

optimization fall into this category, so we propose some additional extensions to handle more general problems.

Constrained optimization. In the literature there are several examples of how to generalize information-theoretic acquisition functions to constrained optimization problems [6, 24, 27, 29, 37, 38, 57, 70]. We propose a simple extension for the JES acquisition function, similar to [70], to handle inequality constraints. In particular, suppose we are interested in maximizing an M-dimensional black-box function $f^{(1:M)}(\mathbf{x})$ subject to K black-box inequality constraints $f^{(M+k)}(\mathbf{x}) \leq 0$ for $k = 1, \ldots, K$. We can model the constraints $f^{(M+k:M+K)}(\mathbf{x})$ as additional independent objectives using the same observation model described in Section 2. The only difference between the constrained setting and the unconstrained setting is the region of integration for the CDF. In the constrained setting, the sampled Pareto front dominates only the vectors satisfying the constraint. Hence the constrained CDF is now of the form $p(\mathbf{z} \in \mathbb{D}^K_{\preceq}(\mathbb{Y}^*))$ where

$$\begin{aligned}
\mathbb{D}^K_{\preceq}(\mathbb{Y}^*) &= \{\mathbf{z} \in \mathbb{R}^{M+K} : (\mathbf{z}^{(1:M)} \preceq \mathbb{Y}^* \text{ and } \mathbf{z}^{(M+1:M+K)} \preceq \mathbf{0}_K) \text{ or } (\mathbf{z}^{(M+1:M+K)} \succ \mathbf{0}_K)\} \\
&= \{\mathbf{z} \in \mathbb{R}^{M+K} : \mathbf{z}^{(1:M+K)} \preceq (\mathbb{Y}^*, \mathbf{0}_K)\} \cup \{\mathbf{z} \in \mathbb{R}^{M+K} : \mathbf{z}^{(1:M+K)} \succ (-\boldsymbol{\infty}_M, \mathbf{0}_K)\} \\
&= \mathbb{D}_{\preceq}((\mathbb{Y}^*, \mathbf{0}_K)) \cup \mathbb{D}_{\succeq}((-\boldsymbol{\infty}_M, \mathbf{0}_K)).
\end{aligned}$$

This region can be decomposed into boxes in the almost same way as before. The only difference is that we now have an additional box arising from region where the constraint is not satisfied. In general, the JES (and MES) acquisition function estimates described here can handle any type of black-box constraint as long as we are able decompose the feasible objective region into boxes. For example, interval constraints of the form $f^{(M+k)}(\mathbf{x}) \in [a, b]$, can also be readily handled in this framework.

Decoupled evaluations. Evaluating all objectives at each iteration can be costly and perhaps unnecessary for practical problems. To address this problem, researchers in BO have considered the use of decoupled [30] acquisition functions $\alpha_{\mathcal{M}}$, which considers the quality of querying a subset of objectives $\{f^{(m)} : m \in \mathcal{M}\}$. To the best of our knowledge, all of the existing decoupled acquisition functions in multi-objective BO are based on information-theoretic acquisition functions [27, 35, 38, 68]. The novel JES-LB2 and MES-LB2 described in this paper possesses this decoupling property because it can be decomposed into a sum of acquisition functions for each objective: $\alpha^{\text{JES-LB2}}(\mathbf{x}|D_n) = \sum_{m=1}^{M} \alpha_m(\mathbf{x}|D_n)$. By Theorem 4.1 of [68], the JES-0 and MES-0 can also be generalized to the decoupled setting via a marginalization argument. Upon reviewing the experimental results of the cited papers, we see that decoupled evaluations typically provide only a marginal improvement over the non-decoupled strategies. This is possibly down to heuristic choice to search the space of subsets. A more principled search method based on some emerging ideas about how to optimize over categorical inputs [28, 52, 60] might be useful to obtain further improvements.

Multi-fidelity Bayesian optimization. For practical optimization problems of interest, it is occasionally possible to evaluate approximations of the true objectives that are much cheaper. This additional degree of freedom has been exploited before in literature under the name of multi-fidelity Bayesian optimization [5, 6, 42, 50, 51, 66, 69, 79, 82]. These strategies have demonstrated some benefit when optimizing with cost constraints on the function evaluations. It is possible to adapt JES to the multi-fidelity by combining the ideas introduced here with the ideas before in the papers referred to above such as using cost weights and conditioning arguments on lower fidelities. The main obstacle to extending this work to the multi-fidelity setting will likely arise from some lengthy algebraic exercises relating to the conditional entropy and box decompositions.