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 [13, 29, 75].

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 [71] with mean

$$\begin{aligned}&\mu_n^{(m)}(X)\\&= \mu_0^{(m)}(X) + \Sigma_0^{(m)}(X, X_n)\left(\Sigma_0^{(m)}(X_n, X_n) + \text{diag}(\sigma^{(m)}(X_n))\right)^{-1}(Y_n^{(m)} - \mu_0^{(m)}(X_n))\end{aligned} \tag{17}$$

and covariance

$$\begin{aligned}&\Sigma_n^{(m)}(X, X)\\&= \Sigma_0^{(m)}(X, X) - \Sigma_0^{(m)}(X, X_n)\left(\Sigma_0^{(m)}(X_n, X_n) + \text{diag}(\sigma^{(m)}(X_n))\right)^{-1}\Sigma_0^{(m)}(X_n, X),\end{aligned} \tag{18}$$

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

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 [36]. To avoid this difficulty, we follow the example of previous work [40, 86], which considers generating approximate samples via random Fourier features [70, 89, 90]. 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 [11] . For the exponential and Matern kernels, this probability distribution is known in closed-form [12, 70]. Assuming $\Sigma_0^{(m)}$ is stationary, an approximate sample of the Gaussian process prior GP($\mu_0^{(m)}, \Sigma_0^{(m)}$) can be written as a Bayesian linear model

$$f_0^{(m)}(\cdot) = \mu_0^{(m)}(\cdot) + \sum_{i=1}^{L} \omega_i \varphi_i(\cdot) \tag{19}$$

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 GP($\mu_n^{(m)}, \Sigma_n^{(m)}$) can be obtained by adding an additional pathwise update via Matherons' rule [89, 90]:

$$f_n^{(m)}(\cdot) = f_0^{(m)}(\cdot) + \sum_{t=1}^{n} \kappa_t \Sigma_0^{(m)}(\cdot, \mathbf{x}_t), \tag{20}$$

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 [90].

B Related work

B.1 Conditional entropy estimates

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 (31) was derived recently in [80] to extend the noiseless MES acquisition function to the multi-objective setting. This extension was called the Pareto frontier entropy search (PFES) and it is equivalent to the acquisition function we call MES-0. An earlier attempt to extend the MES to noiseless setting resulted in the MESMO [4] acquisition function. The MESMO acquisition function is equivalent to the PFES acquisition function if we crudely approximates the dominated space with 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 [80].

The closest work to the lower bound entropy estimate is the GIBBON [59] 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-LB and MES-LB2 can be interpreted as the multi-objective extensions 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 [60, 81]. The MES-MC estimate can be interpreted as the multi-objective extension of these approaches for the single fidelity setting.

B.2 BAX

The JES acquisition function bares some similarity with a special case of the Bayesian Algorithm Execution (BAX) algorithm proposed in [61]. Specifically, if we set $\mathcal{O}_{\mathcal{A}}(f) = (\mathbb{X}^*, \mathbb{Y}^*)$ in [61], then the corresponding BAX acquisition function would reduce to

$$\alpha^{\text{BAX}}(\mathbf{x}|D_n) = H[p(\mathbf{y}|\mathbf{x}, D_n)] - \mathbb{E}_{p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)}[H[p(\mathbf{y}|\mathbf{x}, D_n \cup (\mathbb{X}^*, \mathbb{Y}^*))]]. \tag{21}$$

In the JES acquisition function, we condition on the augmented data set $D_{n*} = D_n \cup (\mathbb{X}^*, \mathbb{Y}^*)$ and the optimality condition $f(\mathbb{X}) \preceq \mathbb{Y}^*$ for the density arising in the conditional entropy term. Whereas in BAX, we only condition on the augmented data set. The BAX acquisition function is a lower bound to JES because conditioning never increases the entropy (Theorem 2.6.5 of [17]).

C Single-objective setting

In this section, we discuss the main differences between the single-objective and multi-objective information-theoretic acquisition functions. At a high level, the only difference between the two settings is whether we use the total ordering over real numbers or the Pareto partial ordering over vectors. Note that in the single-objective setting, $M = 1$, the Pareto partial ordering coincides with the standard total ordering order real numbers, that is to say the definition of the binary relations $\preceq$ and $\prec$ coincides with the standard inequality signs $\leq$ and $<$, respectively. As a result, it is generally possible for multi-objective acquisition functions to be used in the single-objective setting, which is definitely the case for the estimates of our JES acquisition function. We will now comment individually on each information-theoretic acquisition functions on the more subtle differences between the single and multi-objective algorithms.

PES. There are two main difference between the single-objective and multi-objective PES algorithm. Firstly, in the single-objective setting, we sample a single maximizer, $\mathbf{x}^* \sim p(\mathbf{x}^*|D_n)$, based on the standard total ordering, whereas in the multi-objective setting we sample a discrete set of maximizers, $\mathbb{X}^* \sim p(\mathbb{X}^*|D_n)$, based on the Pareto partial ordering. Secondly, the equations governing the expectation propagation updates can be different depending on how the target density is factorised and the modelling assumptions that are made. Therefore, setting $M = 1$ for the multi-objective PES algorithm (also known as PESMO) described in [31] might not exactly recover the same result for a different single-objective implementation of the PES algorithm. In our code we follow the equations proposed in [31], which accounts for single, multi-objective, batch and/or constrained setting.

MES. There are two main difference between the single-objective and multi-objective MES algorithm. Firstly, in the single-objective setting, we sample a single maximum, $y^* \sim p(y^*|D_n)$, based on the standard total ordering, whereas in the multi-objective setting we sample a discrete set of maximums, $\mathbb{Y}^* \sim p(\mathbb{Y}^*|D_n)$, based on the Pareto partial ordering. Secondly, in the single-objective setting the box decomposition is readily available without any effort, $\mathbb{D}_{\preceq}(\{y^*\}) = (-\infty, y^*]$, whereas in the multi-objective setting we have to compute it. The MESMO algorithm [4] and the PFES algorithm [80] both reduce to the single-objective MES algorithm [86] when we set $M = 1$. As mentioned before in Appendix B, the MESMO algorithm is a special case of the PFES algorithm when using a crude approximation to the box decomposition of $\mathbb{D}_{\preceq}(\mathbb{Y}^*)$, which turns out to be exact when there is only one objective.

JES. There are two main difference between the single-objective and multi-objective JES algorithm. Firstly, in the single-objective setting, we sample a single optimal point, $(\mathbf{x}^*, y^*) \sim p((\mathbf{x}^*, y^*)|D_n)$, based on the standard total ordering, whereas in the multi-objective setting we sample a discrete set of optimal points, $(\mathbb{X}^*, \mathbb{Y}^*) \sim p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)$, based on the Pareto partial ordering. Secondly, in the single-objective setting the box decomposition is readily available without any effort, $\mathbb{D}_{\preceq}(\{y^*\}) = (-\infty, y^*]$, whereas in the multi-objective setting we have to compute it. As mentioned before in Section 3, if we do not perform the conditioning step in Algorithm 1, we obtain the MES algorithm.

D Proof of results

Before we begin the proofs, we will restate some important notation. Let $p(\mathbf{y}|\mathbf{x}, D_{n*}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{n*}(\mathbf{x}), \boldsymbol{\Sigma}_{n*}(\mathbf{x}, \mathbf{x}))$ denote the probability density at a point $\mathbf{x} \in \mathbb{X}$ conditional on the data set $D_{n*} = D_n \cup (\mathbb{X}^*, \mathbb{Y}^*)$. Then we denote the m-th standardized function by

$$\gamma_m(z) = \frac{z - \mu_{n*}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x})}} \tag{22}$$

for $m = 1, \dots, M$. For the discrete set of points $\mathbb{Y}^* \subset \mathbb{R}^M$, we decompose the dominated region into J boxes:

$$\mathbb{D}_{\preceq}(\mathbb{Y}^*) = \bigcup_{j=1}^{J} B_j = \bigcup_{j=1}^{J} \prod_{m=1}^{M} (l_j^{(m)}, u_j^{(m)}], \tag{23}$$

where $\mathbf{l}_j = (l_j^{(1)}, \dots, l_j^{(M)})$ are lower bounds and $\mathbf{u}_j = (u_1^{(1)}, \dots, u_J^{(M)})$ are the upper bounds for boxes $j = 1, \dots, J$. Using the box decomposition, we define

$$W_{j,m} = \Phi(\gamma_m(u_j^{(m)})) - \Phi(\gamma_m(l_j^{(m)})) \tag{24}$$

for boxes $j = 1, \dots, J$ and objectives $m = 1, \dots, M$, where Φ is the cumulative distribution function of a standard normal distribution. The first derivative of $W_{j,m}$ (with respect to γ_m) is denoted by

$$G_{j,m} = \phi(\gamma_m(u_j^{(m)})) - \phi(\gamma_m(l_j^{(m)})) \tag{25}$$

and the negative of the second derivative by

$$V_{j,m} = \gamma_m(u_j^{(m)})\phi(\gamma_m(u_j^{(m)})) - \gamma_m(l_j^{(m)})\phi(\gamma_m(l_j^{(m)})), \tag{26}$$

for boxes $j = 1, \dots, J$ and objectives $m = 1, \dots, M$, where ϕ is the probability density function of a standard normal distribution.

D.1 Proof of Proposition 1

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 $\beta \in [0, 1]$.

Proof. The upper bound property follows from the standard result that conditioning on more variables will never increase the entropy (Theorem 2.6.5 of [17]): $H(A|B) \leq H(A)$ 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.

■

D.2 Proof of Lemma 1

Lemma 1. *Let $\mathbb{Y}^* \subset \mathbb{R}^M$ be a finite set and $\mathbf{z} \sim N(\mathbf{a}, \mathit{diag}(\mathbf{b}))$ 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*

$$p(\mathbf{z} \preceq \mathbb{Y}^*) = \sum_{j=1}^J \prod_{m=1}^M \left[\Phi\left(\frac{u_j^{(m)} - a^{(m)}}{\sqrt{b^{(m)}}} \right) - \Phi\left(\frac{l_j^{(m)} - a^{(m)}}{\sqrt{b^{(m)}}} \right) \right]. \tag{27}$$

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

$$\begin{aligned} p(\mathbf{z} \preceq \mathbb{Y}^*) &= \int_{\mathbb{D}_{\preceq}(\mathbb{Y}^*)} p(\mathbf{z}) d\mathbf{z} = \sum_{j=1}^J \int_{B_j} \left(\prod_{m=1}^M p(z^{(m)}) \right) d\mathbf{z} \overset{(1)}{=} \sum_{j=1}^J \prod_{m=1}^M \int_{l_j^{(m)}}^{u_j^{(m)}} p(z^{(m)}) dz^{(m)} \\ &\overset{(2)}{=} \sum_{j=1}^J \prod_{m=1}^M \left[\Phi\left(\frac{u_j^{(m)} - a^{(m)}}{\sqrt{b^{(m)}}} \right) - \Phi\left(\frac{l_j^{(m)} - a^{(m)}}{\sqrt{b^{(m)}}} \right) \right]. \end{aligned}$$

(1) This step follows from the box decomposition of $\mathbb{D}_{\preceq}(\mathbb{Y}^*)$ and the independence between the components of $\mathbf{z}$. (2) This step follows from the definition of the CDF.

■

D.3 Proof of Proposition 2

Proposition 2. *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

$$\text{Cov}\left(y^{(m)}, y^{(m')} \middle| \mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*\right)$$

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

Proof. In the noisy setting, the density of interest is

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

From Lemma 1,

$$p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*}) = \sum_{j=1}^{J} \prod_{m=1}^{M} W_{j,m} = \sum_{j=1}^{J} W_j = W. \tag{28}$$

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 $p(\mathbf{y}|\mathbf{x}, D_{n+})$. By standard Gaussian conditioning $p(\mathbf{y}|\mathbf{x}, D_{n+}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{n+}(\mathbf{x}), \Sigma_{n+}^{(m)}(\mathbf{x}, \mathbf{x}))$ 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 $m = 1, \dots, M$. Using the box decomposition, the posterior CDF is equal to

$$p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n+}) = \sum_{j=1}^{J} \prod_{m=1}^{M} \left[\Phi\left(\frac{u_j^{(m)} - \mu_{n+}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n+}^{(m)}(\mathbf{x}, \mathbf{x})}} \right) - \Phi\left(\frac{l_j^{(m)} - \mu_{n+}^{(m)}(\mathbf{x})}{\sqrt{\Sigma_{n+}^{(m)}(\mathbf{x}, \mathbf{x})}} \right) \right]. \tag{29}$$

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 $\bar{y}^{(m)} = (y^{(m)} - \mu_{n*}^{(m)}(\mathbf{x})) / \sqrt{\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x}) + \sigma^{(m)}(\mathbf{x})}$ is the standardized observation, which is distributed according to a standard normal random variable for objectives $m = 1, \dots, M$. 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 $\bar{\mathbf{y}}$.

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

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,

$$\int_{\mathbb{R}} \left[\Phi(\gamma_m^+(u_j^{(m)})) - \Phi(\gamma_m^+(l_j^{(m)})) \right] \phi(\bar{y}^{(m)} - t^{(m)}) d\bar{y}^{(m)}$$

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

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 $\mathbf{t} = \mathbf{0}_M$.

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

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 $m = m'$,

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

The moments of $y^{(m)}$ can be now derived by reversing the initial linear transformation: $y^{(m)} = \bar{y}^{(m)} \sqrt{\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x}) + \sigma^{(m)}(\mathbf{x})} + \mu_{n*}^{(m)}(\mathbf{x})$. The first moment is

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

The second moment:

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

This implies

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

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

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

and for $m = m'$

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

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

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

and the variance is

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

■

D.4 Proof of Proposition 3

Proposition 3. *The information-theoretic acquisition functions* α^{PES}, α^{MES} *and* α^{JES} *are invariant to reparameterization 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.*

Proof. 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.

■

E Noiseless entropy estimate

In this section, we present the conditional entropy estimate for the zero observation variance setting and an ad hoc extension for noisy setting. Firstly, if we assume the observation variance is zero, the conditional distribution of interest is a truncated multivariate normal:

$$p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*) = \frac{p(\mathbf{y}|\mathbf{x}, D_{n*})}{p(f(\mathbf{x}) \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*})}\mathbb{I}[\mathbf{y} \preceq \mathbb{Y}^*], \tag{30}$$

which is known to have the following analytical equation for the entropy:

Proposition 4. *(Theorem 3.1. in [80]) 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*

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

Proof. See [80] for the proof of this result.

■

Assuming $\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{0}_M$, we have

$$\begin{aligned}\alpha^{\text{JES-0}}(\mathbf{x}|D_n) &= H[p(\mathbf{y}|\mathbf{x}, D_n)] - \mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}[H[p(\mathbf{y}|\mathbf{x}, D_{n*}, f(\mathbf{x}) \preceq \mathbb{Y}^*)]]\\ &= -\mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}[\log(W)]\\ &\quad + \frac{1}{2}\sum_{m=1}^{M}(\log(\Sigma_n^{(m)}(\mathbf{x},\mathbf{x})) - \mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}[\log(\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}))])\\ &\quad + \mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}\left[\frac{1}{2W}\sum_{j=1}^{J} W_j \sum_{m=1}^{M}\frac{V_{j,m}}{W_{j,m}}\right].\end{aligned}$$

The first term is equal to expectation of the negative log-probability of the noiseless observation lying below Pareto front:

$$-\mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}[\log(W)] = -\mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}[\log(p(\mathbf{y} \preceq \mathbb{Y}^*|\mathbf{x}, D_{n*}))].$$

This term accounts for the exploitation because it puts more emphasis on the points that are likely to lie above the Pareto front. The remaining terms account for the exploration by placing more emphasis on reducing the uncertainty at input location. To still take advantage of the result in Proposition 4 for the noisy observation setting, we propose an ad hoc extension that adjusts the exploration term to include the effects of the observation noise. Specifically, we propose a modification which replaces the difference in log variances in the exploration term,

$$\log(\Sigma_n^{(m)}(\mathbf{x},\mathbf{x})) - \mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}[\log(\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}))]$$

with the differences in log variances plus observation noise,

$$\log(\Sigma_n^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})) - \mathbb{E}_{p((\mathbb{X}^*,\mathbb{Y}^*)|D_n)}[\log(\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})))],$$

where $\sigma^{(m)}(\mathbf{x}))(\mathbf{x})$ is the observation variance for objectives $m = 1, \dots, M$ at $\mathbf{x} \in \mathbb{X}$. Using this adjustment, we define the resulting conditional entropy estimate by

$$\begin{aligned}&h^{\text{JES-0}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n)\\ &= \frac{M}{2}\log(2\pi e) + \frac{1}{2}\sum_{m=1}^{M}\log(\Sigma_{n*}^{(m)}(\mathbf{x},\mathbf{x}) + \sigma^{(m)}(\mathbf{x})) + \log W - \frac{1}{2W}\sum_{j=1}^{J} W_j \sum_{m=1}^{M}\frac{V_{j,m}}{W_{j,m}}.\end{aligned} \tag{31}$$

Empirically, we observe that the performance of this conditional entropy estimate is in-line with the other conditional entropy approximations.

F 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 (28), whilst the CDF in the numerator is

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

from (29). By sampling $\mathbf{y}_i \sim p(\mathbf{y}|\mathbf{x}, D_{n*})$ for $i = 1, \dots, I$, we can approximate the entropy with the following Monte Carlo average:

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

where the Monte Carlo entropy estimate is given by

$$h^{\text{JES-MC}}((\mathbb{X}^*, \mathbb{Y}^*); \mathbf{x}, D_n) = -\frac{1}{WI}\sum_{i=1}^{I} W^+(\mathbf{y}_i) \log(W^+(\mathbf{y}_i) p(\mathbf{y}_i|\mathbf{x}, D_{n*})) + \log(W). \tag{32}$$

Instead of generating new samples for each call of the acquisition function, we follow the general wisdom in BO [88] and apply the reparameterization 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,

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

where the expectations are known,

$$\mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}[W^+(\mathbf{y})] = W, \tag{33}$$

$$-\mathbb{E}_{p(\mathbf{y}|\mathbf{x}, D_{n*})}[\log(p(\mathbf{y}|\mathbf{x}, D_{n*})] = \frac{M}{2}\log(2\pi e) + \frac{1}{2}\sum_{m=1}^{M} \log(\Sigma_{n*}^{(m)}(\mathbf{x}, \mathbf{x}) + \sigma^{(m)}(\mathbf{x})) \tag{34}$$

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 reparameterization trick worked reasonably well out of the box. One word of caution [43]: combining both control variates and quasi-Monte Carlo could lead to an increase in variance if the coefficients are naively estimated.

G 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 [53]. 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 [54]. Similar to [59], 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

$$
\zeta(\mathbf{x}^{[i]}) = \frac{M}{2q}\log(2\pi e) - \mathbb{E}_{p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)}\left[H[p(\mathbf{y}^{[i]}|\mathbf{x}^{[i]}, D_{n*}, f(\mathbb{X}^*) \preceq \mathbb{Y}^*)]\right]. \tag{35}
$$

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 [54]. 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 (35) is replaced by the corresponding Monte Carlo estimate. The submodularity derivation above continues to hold when using the conditional entropy approximations.

■

H 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 [89]. An evaluation of a sample at a set of inputs $X \subset \mathbb{X}$ has a linear cost depending on $|X|$ [89].

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 [22]. 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 [22]. 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 [55] 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 (18) 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 [30], which computes the approximate log-determinant, which only uses a linear cost of $O(q)$—this approximation approach is outlined in [25].

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 $h^{\text{JES-0}}$ (31) and $h^{\text{JES-LB2}}$ (14) comes from the evaluation of the variance. For $h^{\text{JES-LB}}$ (13), we additionally have to populate an $M \times M$ matrix and compute its log-determinant. For $h^{\text{JES-MC}}$ (32), we need to initialize a set of I base samples for the reparameterization trick [88] described in Appendix F—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 [31] consider using expectation propagation [58]. 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 param-

eters 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 [31] to approximate the PES acquisition. The batch extension is derived in a follow-up work [33]. 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}.

I Estimation error

Overall, there are five sources of estimation error when approximating the batch JES acquisition function. In this section, we briefly enumerate and discuss these errors below.

1. The first source of error arises from replacing the global optimality condition with the local optimality condition. This turns out to be a lower bound approximation:

$$\begin{aligned}
\alpha^{\text{JES}}(\mathbf{x}|D_n) &= H[p(\mathbf{y}|\mathbf{x}, D_n)] - \mathbb{E}_{p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)}[H[p(\mathbf{y}|\mathbf{x}, D_n, (\mathbb{X}^*, \mathbb{Y}^*))]] \\
&= H[p(\mathbf{y}|\mathbf{x}, D_n)] - \mathbb{E}_{p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)}[H[p(\mathbf{y}|\mathbf{x}, D_n \cup (\mathbb{X}^*, \mathbb{Y}^*), f(\mathbb{X}) \preceq \mathbb{Y}^*)]] \\
&\geq H[p(\mathbf{y}|\mathbf{x}, D_n)] - \mathbb{E}_{p((\mathbb{X}^*, \mathbb{Y}^*)|D_n)}[H[p(\mathbf{y}|\mathbf{x}, D_n \cup (\mathbb{X}^*, \mathbb{Y}^*), f(\mathbf{x}) \preceq \mathbb{Y}^*)]],
\end{aligned}$$

because conditioning on more variables will never increase the entropy. The error from this lower bound approximation appears also in the majority of the work on entropy based acquisition functions. Empirically, we observed that this approximation leads to a very minor change to the selected point for the sequential acquisition function. The effects of this error on the batch acquisition function is unclear because computing an unbiased estimate of the exact batch acquisition function over the whole input space is too expensive to do in practice.

2. The second and third source of error arises from the Monte Carlo approximation of the intractable expectation and the discrete approximation of the Pareto optimal points, respectively. The error of this step could be reduced by increasing the number of samples or optimal points at the expense of more computation. In the experiments we conducted, we set the number of Monte Carlo samples as $S = 10$ and the number of optimal points as $p = 10$. This combination seemed to work well under our collection of problems and the sensitivity analysis we conducted in Appendix L.3 seems to indicate that there is a diminishing gain in performance when we increase the number of samples or optimal points further.
3. The fourth source of error comes from estimating the conditional entropy. This can be computed exactly in the noiseless setting, but has to be estimated in the other settings. The Monte Carlo estimate gives an unbiased estimate of this term, which can be made more accurate by increasing the number of samples. In the experiments presented in Appendix L, we observed very little difference in the points that are selected when using the different conditional entropy estimates. As a result, we recommend using the cheapest estimates which in most cases is the one based on the moment-matching approach.
4. The fifth source of error comes from estimating the batch acquisition function using the lower bound. This error is only present when the batch size is greater than one: $q > 1$. Quantifying this error is hard to do empirically because the exact batch acquisition function is too expensive to estimate unbiasedly. Nevertheless, the lower bound batch acquisition function and its approximation are still principled acquisition functions because they can be written as determinantal point processes (DPPs) [54]. This property was derived and used in the proof of submodularity in Appendix G.

   As discussed in the BO literature [15, 49, 87], batch acquisition functions based on DPPs are powerful because they can promote diverse batches in high-quality regions. The trade-off between diversity and quality (Section 3.1 in [54]) is evident in the form of the DPP kernel, $K^{(m)}_{i,j} = q_i q_j A^{(m)}_{i,j}$, where $q_i = \exp(\zeta_i/M)$ can be interpreted as the quality of item i, whilst $A^{(m)}_{i,j}$ corresponds to a notion of similarity. In the approximation, we use a Monte Carlo estimate for the term ζ_i, which only arises when computing the quality term. As a result, the batch that is selected with this approximation might differ slightly from the optimal batch but the diversity of the batch will still be high because of the matrix $A^{(m)}$.

J 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 8, Figure 9 and Figure 10, respectively. To obtain the ground truth for the acquisition function we run the rejection sampling scheme discussed in [40]. 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.

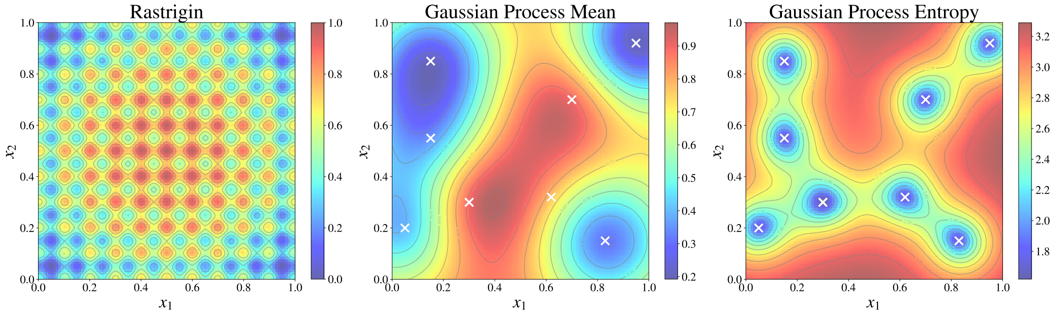


Figure 7: The contours for the Gaussian process posterior mean and entropy after making 8 noisy observations of the (normalized) Rastrigin objective function ($d = 2, M = 1$).

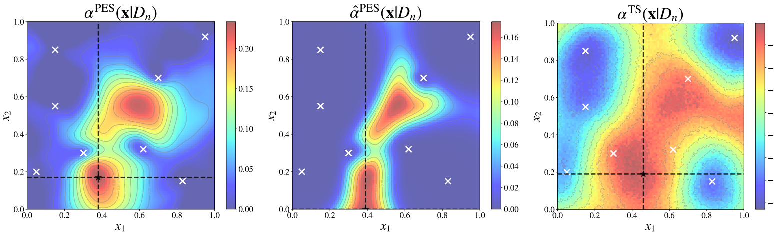


Figure 8: The contours for the predictive entropy search acquisition and its approximation obtained via expectation propagation. For reference we also include a random Thompson sample. The location of the maximizer is highlighted using a pair of dotted black lines.

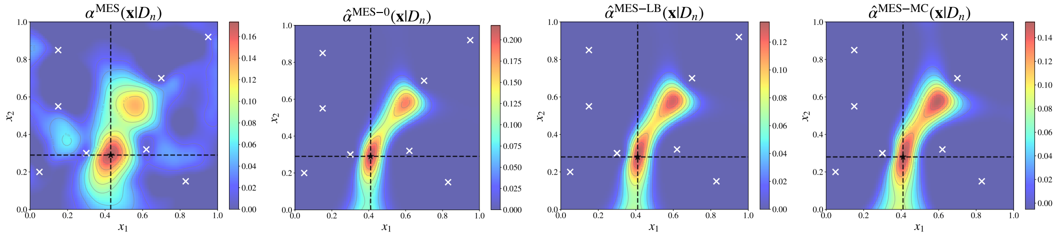


Figure 9: The contours for the maximum value entropy search acquisition function and its approximation obtained via the zero noise approximation, moment matching and Monte Carlo. The location of the maximizer is highlighted using a pair of dotted black lines.

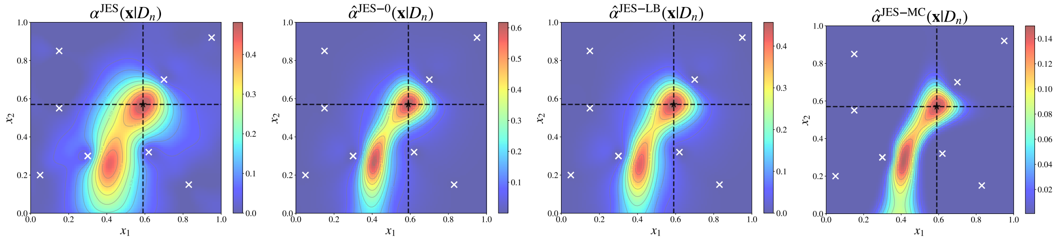


Figure 10: The contours for the joint entropy search acquisition function and its approximation obtained via the zero noise approximation, moment matching and Monte Carlo. The location of the maximizer is highlighted using a pair of dotted black lines.

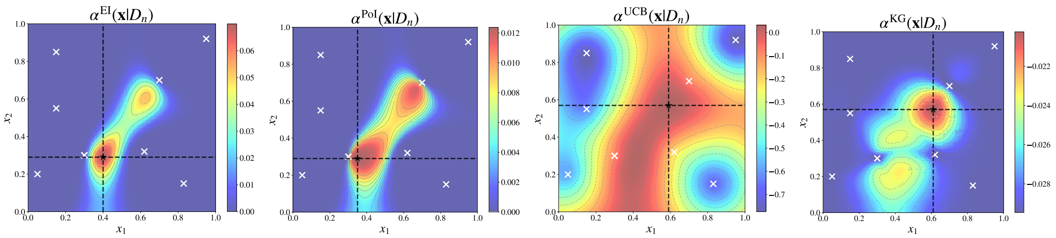


Figure 11: The contours for the expected improvement, probability of improvement, upper confidence bound ($\beta = 3$) and knowledge gradient acquisition function. The location of the maximizer is highlighted using a pair of dotted black lines.

K 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 parameterizations 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 parameterizations. 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 [23, 94]. 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:

$$U_{\text{HV}}(A) = \frac{\pi^{M/2}}{2^M \Gamma(M/2+1)} \mathbb{E}_{p(\boldsymbol{\lambda})} \left[\max_{\mathbf{a} \in A} s_{\boldsymbol{\lambda}}(\mathbf{a}) \right] \tag{36}$$

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

$$s_{\boldsymbol{\lambda}}(\mathbf{a}) = \min_{m=1,\dots,M} \left(\max \left(0, (a^{(m)} - r^{(m)})/\lambda^{(m)} \right) \right)^M . \tag{37}$$

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

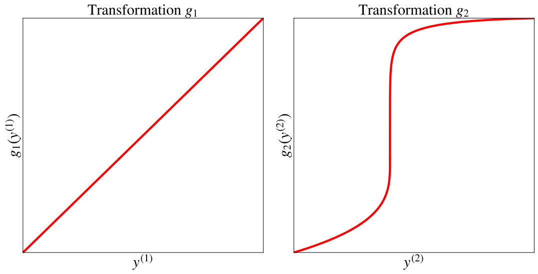
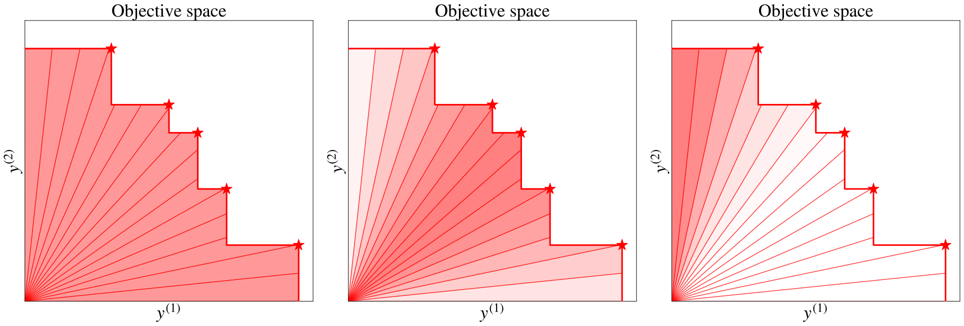


Figure 12: The transformation functions g_1 and g_2 used in Figure 4b.



(a) $\text{Beta}(1,1)$ (b) $\text{Beta}(2.5,2.5)$ (c) $\text{Beta}(5,2)$

Figure 13: 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

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

Consequently, 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 $\text{Beta}(a^{(m)}, b^{(m)})$ to generate samples from $\mathbf{w} \in [0,1]^{M-1}$. In Figure 13, we present the radial contributions to the GHV for different weight distributions $p(\mathbf{w})$.

To isolate performance in one particular radial region, we propose the use of a uniform distribution over some subset of $[0,1]^{M-1}$. For example in Figure 6, we chose three different subsets of $[0,1]$ to isolate three different regions of the two-dimensional objective space.

Remark. We are not the first to observe the importance of weighting the HV towards regions of interest. For example, an early paper [99] 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.

L Experiments

L.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) [30], NumPy (1.21.2), Pymoo (0.5.0) [10] [37], PyTorch (1.9.0) [66] and SciPy (1.7.3) [85]. 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/benmltu/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 [85]. We use the exact gradient inferred using automatic differentiation [66] 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 [31]. 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 [22], 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 [18]. 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 L.9, 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 [55]. The box decompositions were performed in sequence instead of being executed in parallel. From the wall times presented in Appendix L.9, 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) [12] 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 [12], 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 use the same modification for the batch extension.
- **ParEGO/ NParEGO.** The random scalarization strategy (ParEGO) [51] and its noisy counterpart (NParEGO) [19] 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) [18] and its noisy counterpart (NEHVI) [19] 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 [18]. For the batch setting, we considered a sequentially greedy optimization strategy.
- **PES.** The predictive entropy search (PES) [31, 33, 39, 40] acquisition function is approximated using expectation propagation. We implemented this algorithm from scratch in BoTorch under the guidance of the supplementary material in [31]. 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 [31]. This extension appears to be equivalent to the approach described in [33]. We optimized the resulting acquisition function using a joint optimization approach.
- **MES.** The maximum value entropy search (MES) [80, 86] 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 [18, 19]. For the batch setting, we consider the lower bound described by (15). We optimized the resulting acquisition function using a greedy optimization approach.
- **JES.** The joint entropy search (JES) [80, 86] 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 (15). 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 $N_{\text{off}} = 10$.

L.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 [97] 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 [27] 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 [45]. Unlike the original paper [27], 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 [56] 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 [56]. We used the open-source implementation available in BoTorch [3, 20]. 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 [65, 73, 83] 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 [65, 73, 83]. 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 $\mathbf{r} = (250, -20000, -25000, -15.0)$.

L.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, 31, 80, 86]. 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 mean log HV discrepancy with two standard errors over the runs in Figure 14 and Figure 15. We report the wall times in Figure 16.

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.

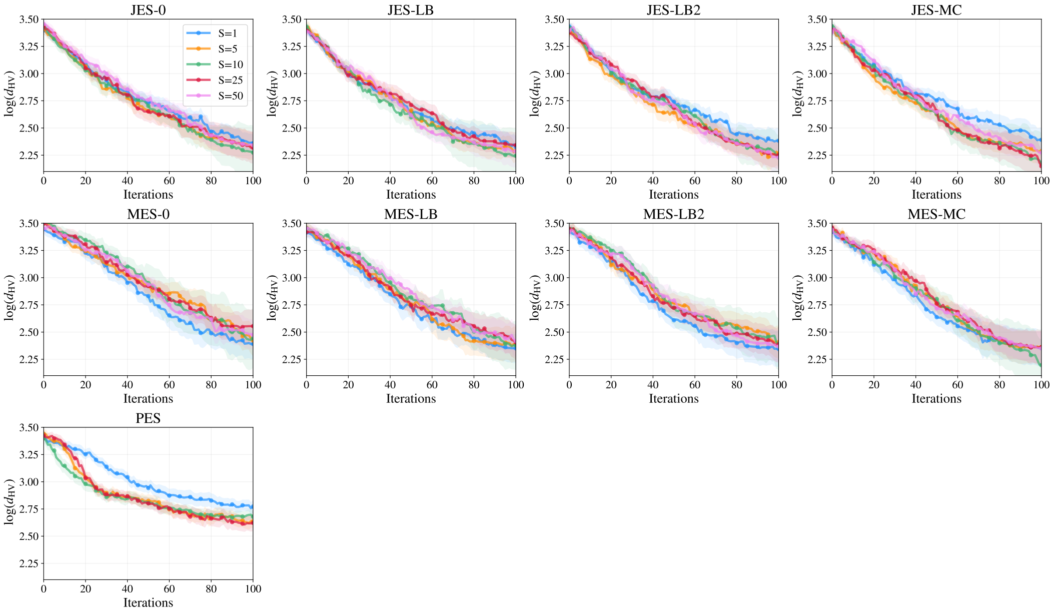


Figure 14: A comparison of the mean logarithm HV discrepancy with two standard errors over different number of Monte Carlo samples S.

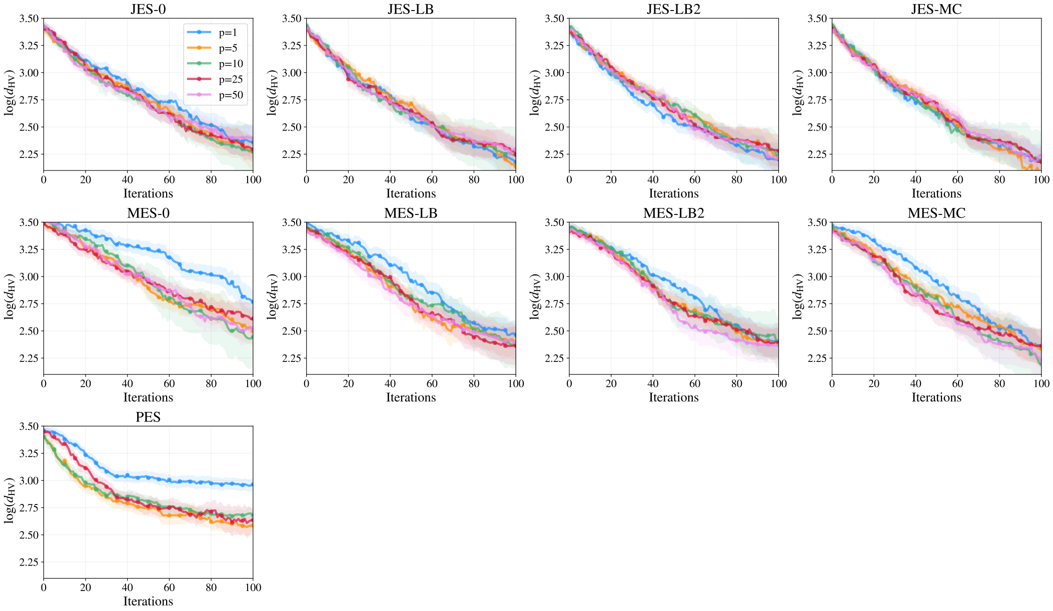


Figure 15: A comparison of the mean logarithm HV discrepancy with two standard errors over different number of Pareto optimal samples p.

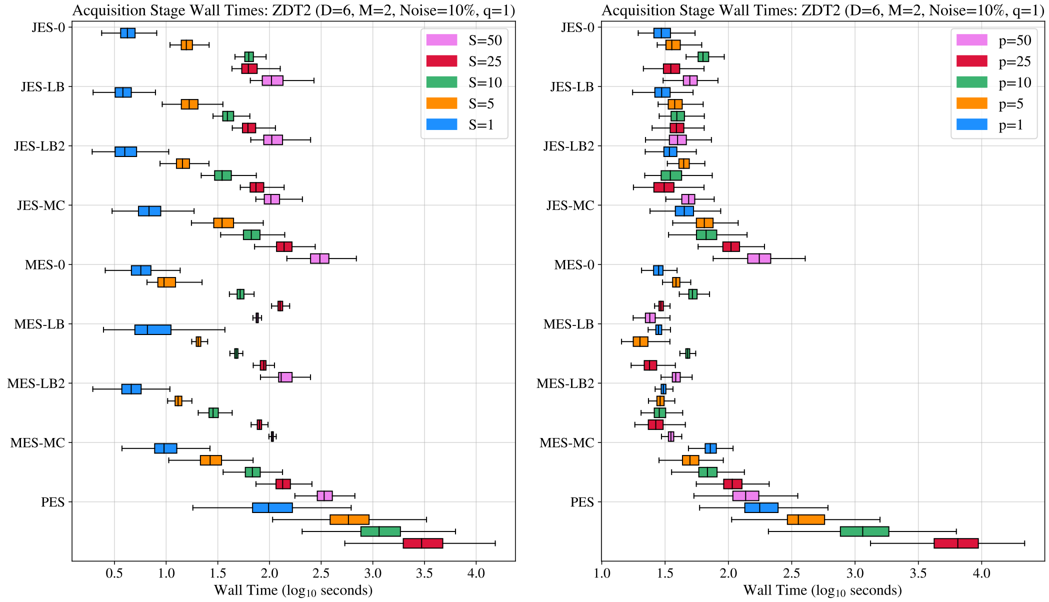


Figure 16: A box-plot comparison of the wall times for the acquisition stage over different number of Monte Carlo samples (left) and the number of Pareto optimal samples (right). The box-plots include the median, interquartile range and the extreme values after excluding the outliers. The acquisition 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.

L.4 Noise levels

In the section we empirically analyse how sensitive the approximations to the information-theoretic acquisition functions are when we increase noise levels. In Figure 17 and Figure 18, we plot the mean logarithm HV discrepancy for the JES and MES estimates on the ZDT2 (D=2, M=2, q=1) benchmark when the recommended points are obtained by maximizing the posterior mean. In both examples, we observe that the performance decreases as the noise levels increases. On the whole the JES estimates perform very similarly across the board. For the MES results, we see that the MES-0 estimate is noticeable weaker than the rest even after the ad hoc correction described in Appendix E.

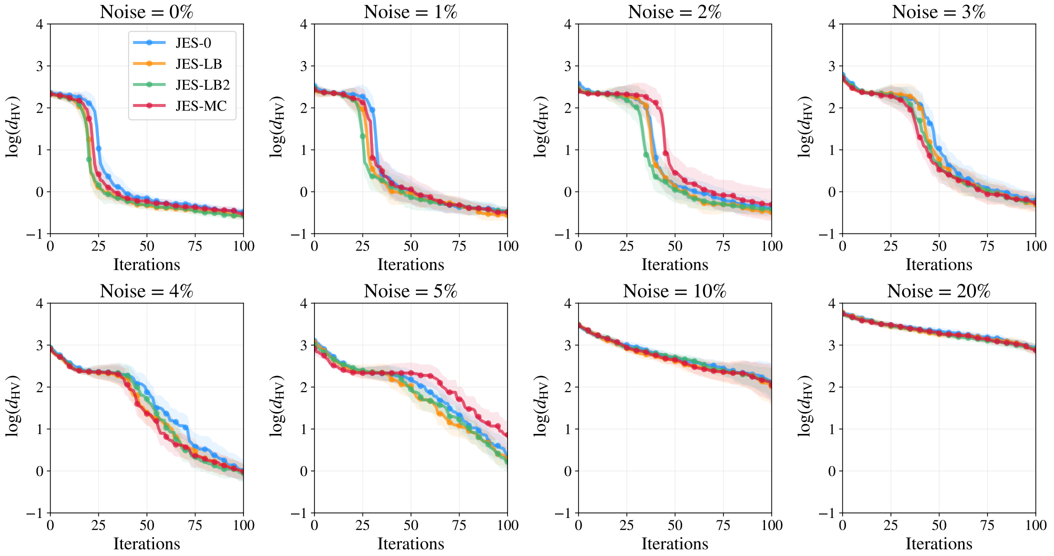


Figure 17: A comparison of the mean logarithm HV discrepancy with two standard errors for different noise levels. The recommended set of points were obtained by maximizing the posterior mean.

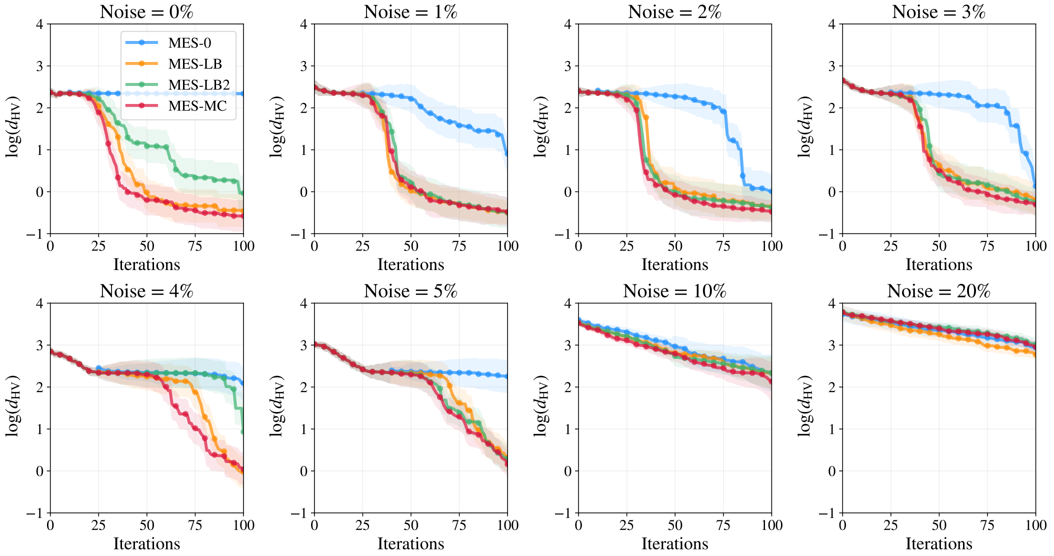


Figure 18: A comparison of the mean logarithm HV discrepancy with two standard errors for different noise levels. The recommended set of points were obtained by maximizing the posterior mean.

L.5 In-sample results

In this section we present the results of the standard hypervolume when we restrict the recommended Pareto set $\hat{\mathbb{X}}^*$ to be a subset of the sampled locations: $\hat{\mathbb{X}}^* = \arg\max_{\mathbf{x} \in X_N} \boldsymbol{\mu}_N(\mathbf{x})$. The complete results are presented in Figure 19. Our main finding is that information-theoretic strategies have a tendency to not directly query the best performing points but instead opt for more informative points that will reduce the overall model uncertainty over the optimal points. As a result of this behaviour, information-theoretic algorithms tend to perform well when we assess the Pareto set over the whole input space $\mathbb{X}$ and less so when we only assess the performance over the sampled locations X_N. If directly querying high-performing points is important, it might be advantageous to use an epsilon greedy strategy, where points are occasionally picked greedily according to the posterior mean.

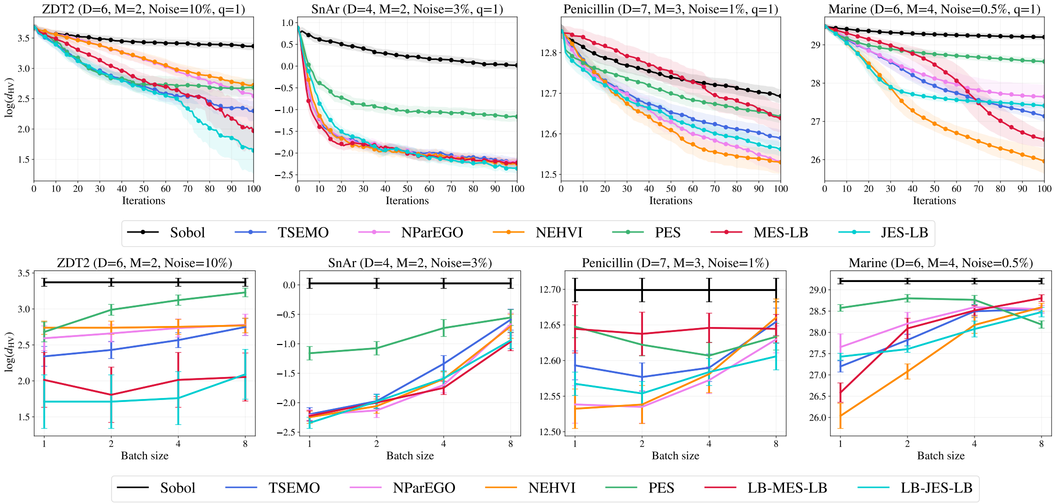


Figure 19: A comparison of the mean logarithm HV discrepancy with two standard errors over one hundred runs for the four benchmark problems on a subset of the algorithms when we restrict the approximate Pareto set to the sampled points. On the top, we present the results for the sequential experiment, whilst on the bottom we present the final results on the batch experiments with different batch sizes.

L.6 Comparing the entropy estimates

In this section we compare the results using the different conditional entropy estimates for the different experiments. In Figure 20 and Figure 21 we present the results for the JES and MES estimates respectively. We observed that the different conditional entropy estimates seem to perform similarly across the board. As a result, we advocate the use of the cheapest estimate, which are typically the lower bound estimates. On the Marine experiment, we observed that the zero-variance estimate was noticeably weaker even after we applied the ad hoc correction described in Appendix E. Therefore, we generally recommend against using the zero-variance estimate if possible.

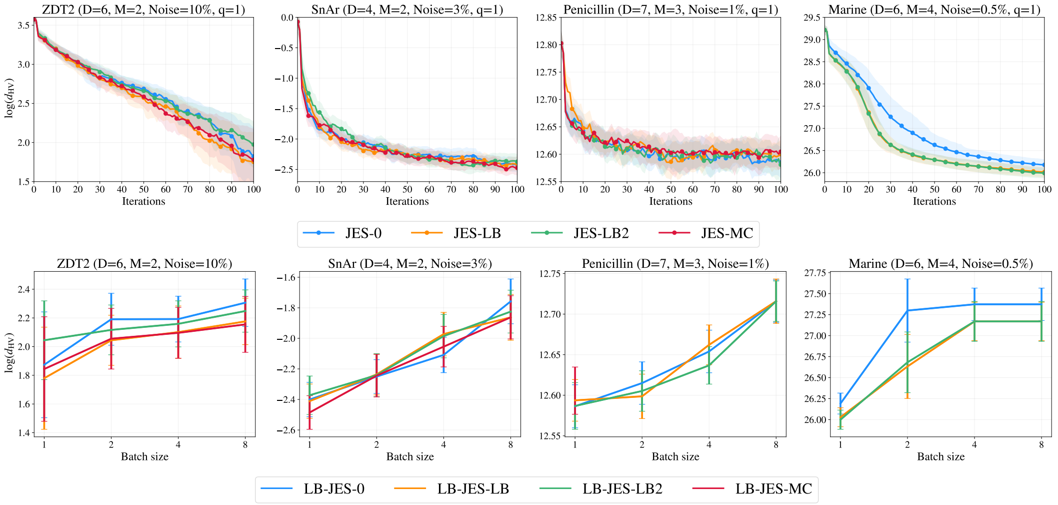


Figure 20: A comparison of the mean logarithm HV discrepancy with two standard errors over one hundred runs for the four benchmark problems for the JES algorithms. On the top, we present the results for the sequential experiment, whilst on the bottom we present the final results on the batch experiments with different batch sizes.

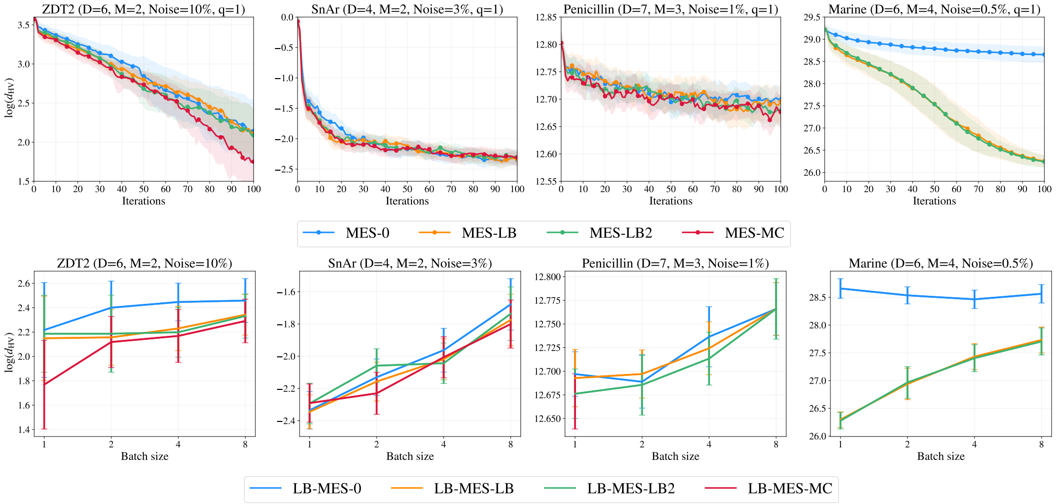


Figure 21: A comparison of the mean logarithm HV discrepancy with two standard errors over one hundred runs for the four benchmark problems for the MES algorithms. On the top, we present the results for the sequential experiment, whilst on the bottom we present the final results on the batch experiments with different batch sizes.

L.7 Comparing the improvement-based algorithms

In this section we present the results for the improvement-based algorithms: NParEGO, ParEGO, EHVI and NEHVI. On the whole, we observe that the greedier strategy which ignores the noise seems to perform reasonably well compared to the strategy which accounts for the noise.

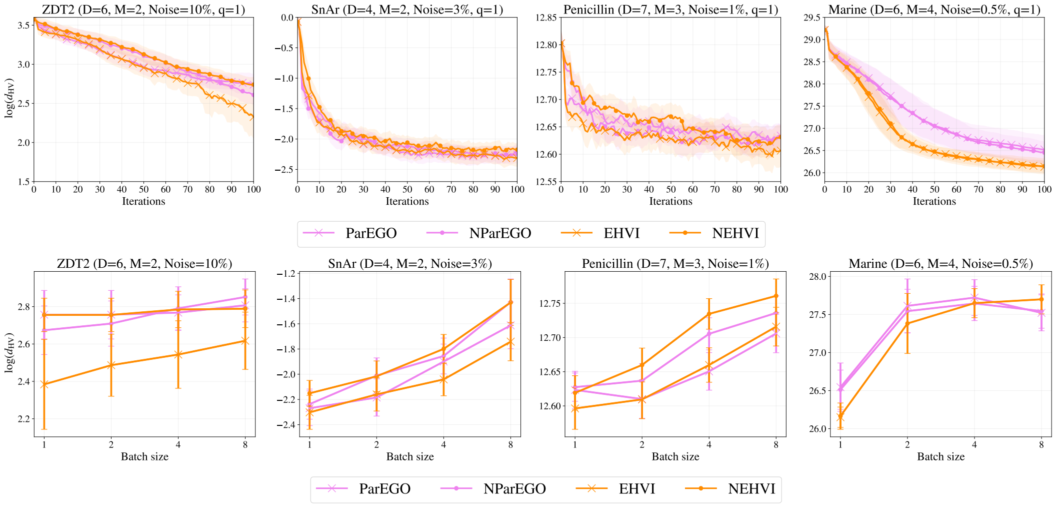


Figure 22: A comparison of the mean logarithm HV discrepancy with two standard errors over one hundred runs for the four benchmark problems for the improvement-based algorithms. On the top, we present the results for the sequential experiment, whilst on the bottom we present the final results on the batch experiments with different batch sizes.

L.8 Generalized hypervolume

In this section we produce profile plots of the generalized hypervolume at the final time instance. We only consider the median performance from the multiple runs when setting the approximation set to be the maximum of the final posterior mean: $\hat{\mathbb{X}}^* = \arg\max_{\mathbf{x}\in\mathbb{X}} \boldsymbol{\mu}_N(\mathbf{x})$

L.8.1 Weight distributions

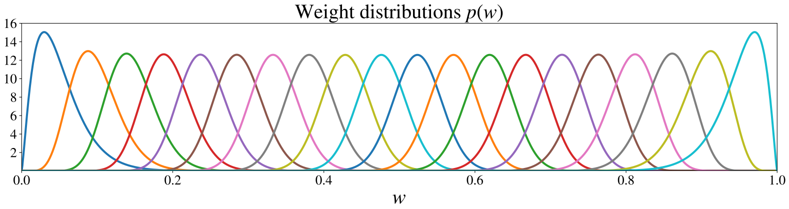


Figure 23: The Beta distributions used to generate the weights for the logarithm GHV discrepancy.

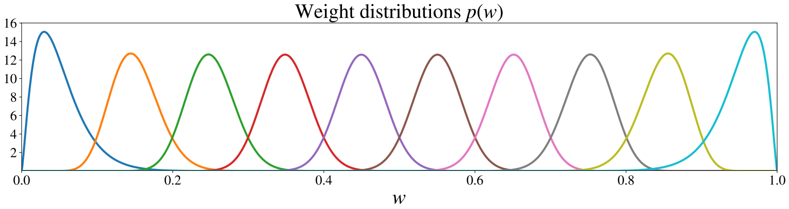


Figure 24: The Beta distributions used to generate the weights for the logarithm GHV discrepancy.

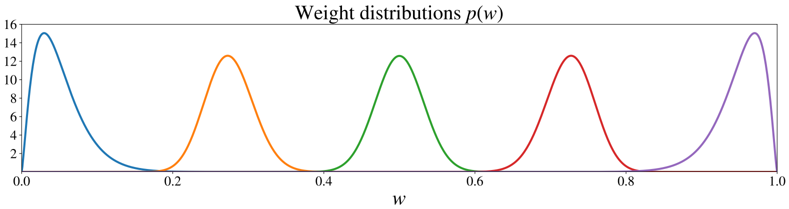


Figure 25: The Beta distributions used to generate the weights for the logarithm GHV discrepancy.

L.8.2 ZDT2(D=2, M=2, Noise=10%, q=1)

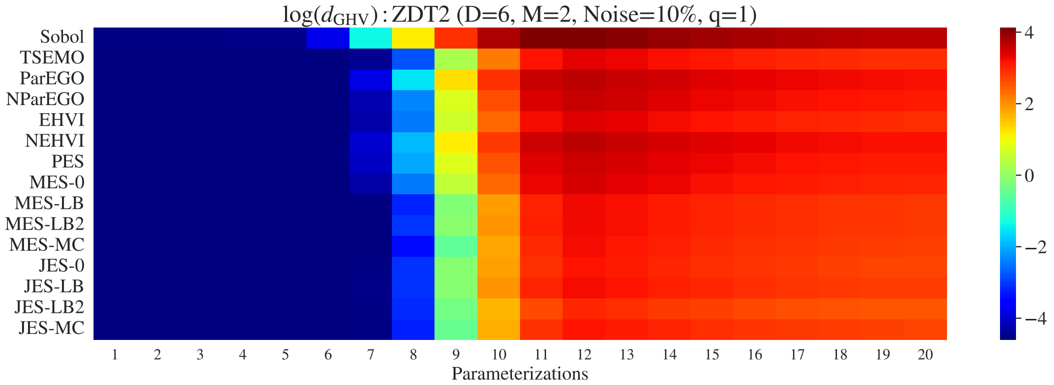


Figure 26: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 23.

L.8.3 SnAr (D=4, M=2, Noise=3%, q=1)

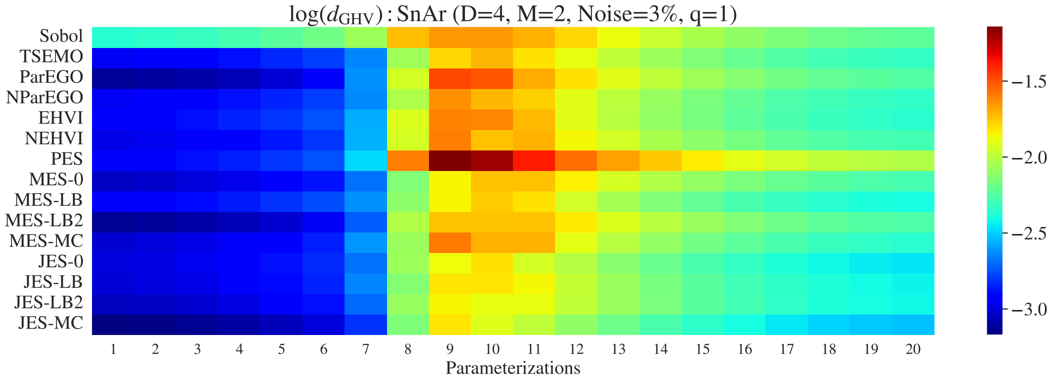


Figure 27: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 23.

L.8.4 Penicillin (D=7, M=3, Noise=1%, q=1)

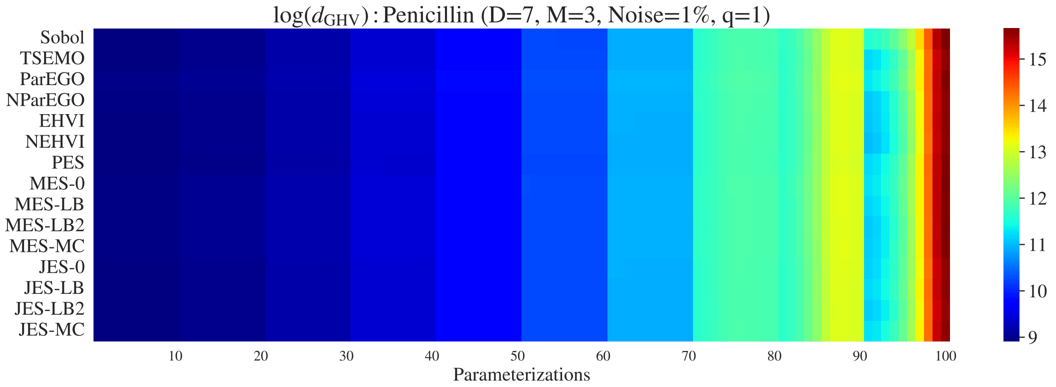


Figure 28: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 24.

L.8.5 Marine (D=6, M=4, Noise=0.5%, q=1)

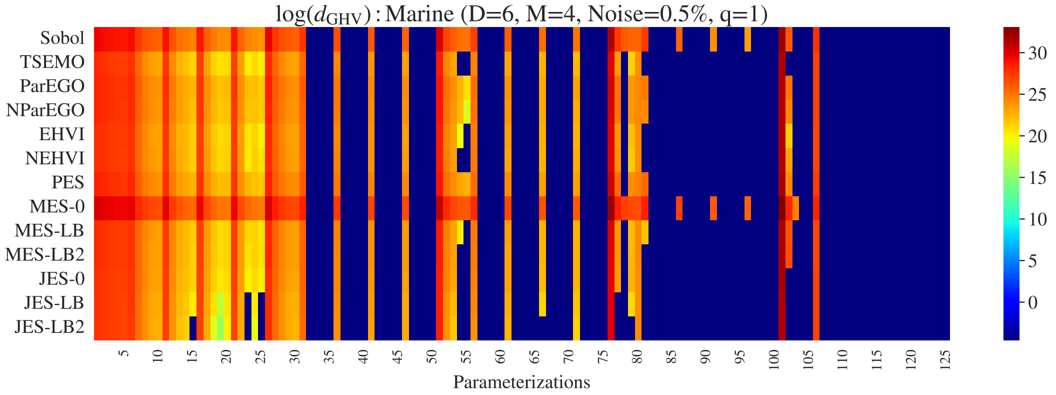
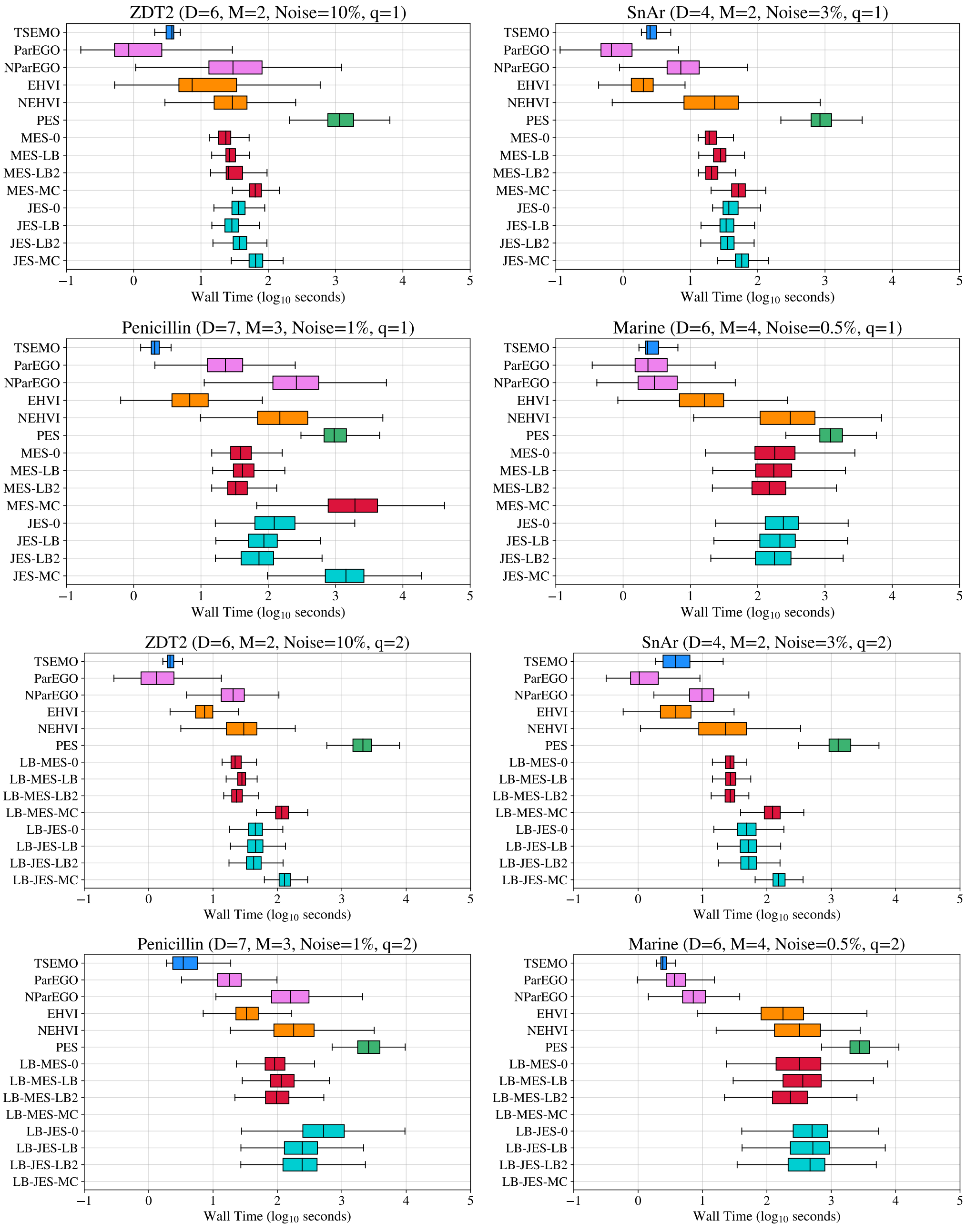


Figure 29: A heat map comparison of the final median logarithm GHV discrepancy. The weight distributions are described in Figure 25.

L.9 Wall times


ZDT2 (D=6, M=2, Noise=10%, q=1)
SnAr (D=4, M=2, Noise=3%, q=1)
TSEMO
ParEGO
NParEGO
EHVI
NEHVI
PES
MES-0
MES-LB
MES-LB2
MES-MC
JES-0
JES-LB
JES-LB2
JES-MC
−1
0
1
2
3
4
5
Wall Time ($\log_{10}$ seconds)
Penicillin (D=7, M=3, Noise=1%, q=1)
Marine (D=6, M=4, Noise=0.5%, q=1)
ZDT2 (D=6, M=2, Noise=10%, q=2)
SnAr (D=4, M=2, Noise=3%, q=2)
LB-MES-0
LB-MES-LB
LB-MES-LB2
LB-MES-MC
LB-JES-0
LB-JES-LB
LB-JES-LB2
LB-JES-MC
Penicillin (D=7, M=3, Noise=1%, q=2)
Marine (D=6, M=4, Noise=0.5%, q=2)

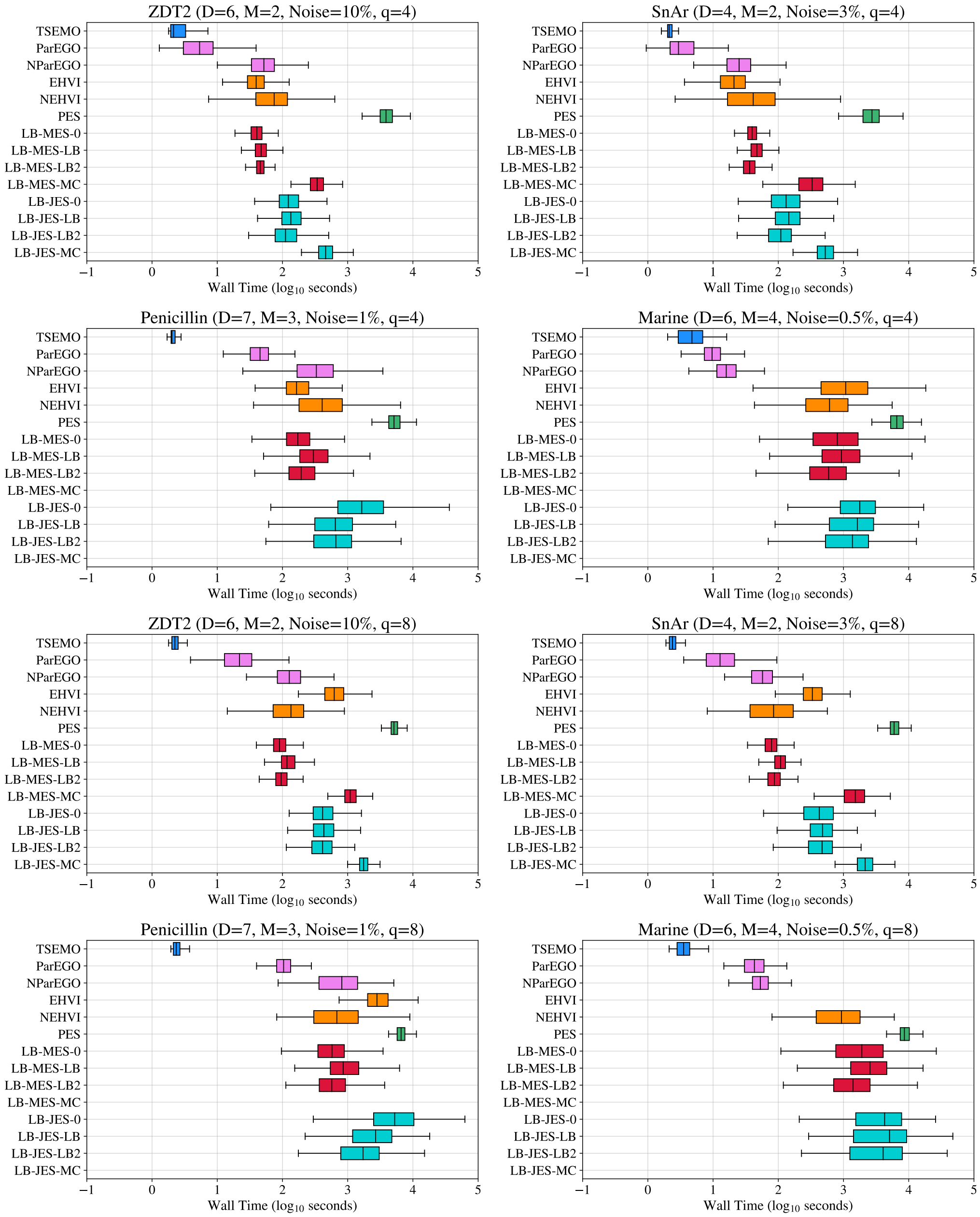
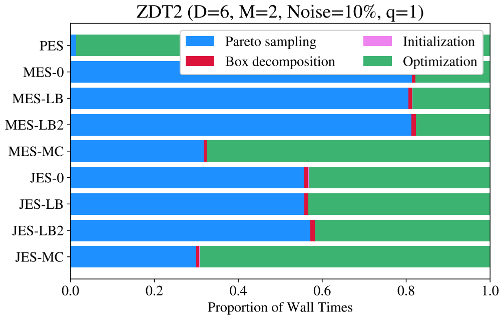
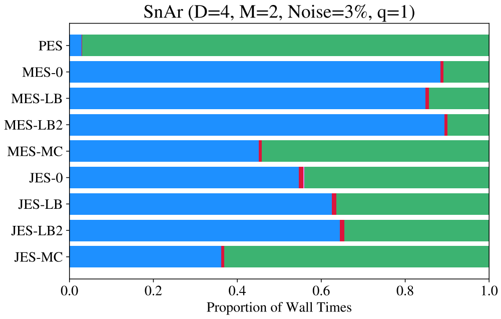
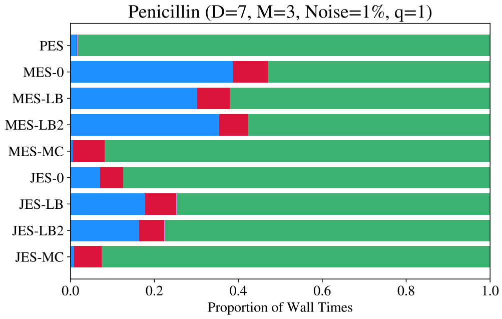
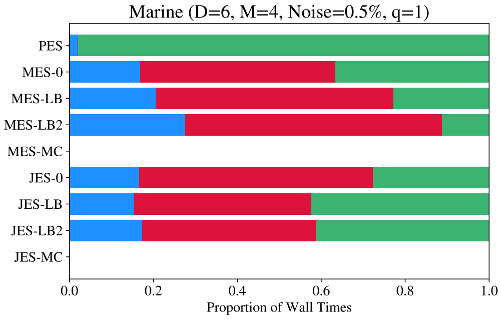


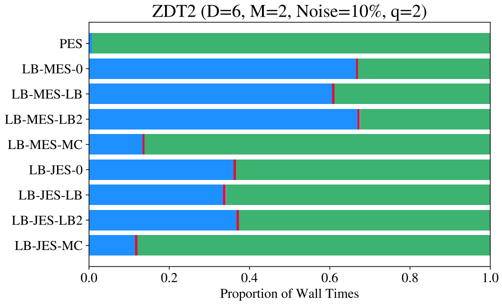
Figure 29: A box-plot comparison of the wall times for the acquisition stage, which includes the median, interquartile range and the extreme values after excluding the outliers. The acquisition stage includes any initialization computations such as the box decompositions and sampling the Pareto optimal 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.

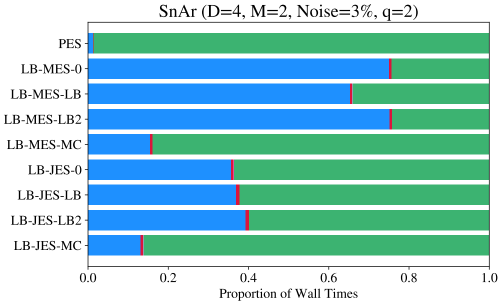

ZDT2 (D=6, M=2, Noise=10%, q=1)
Pareto sampling
Box decomposition
Initialization
Optimization
PES
MES-0
MES-LB
MES-LB2
MES-MC
JES-0
JES-LB
JES-LB2
JES-MC
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of Wall Times

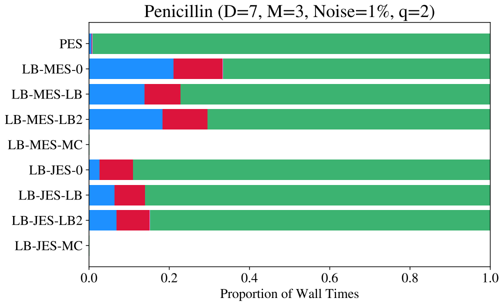

SnAr (D=4, M=2, Noise=3%, q=1)
PES
MES-0
MES-LB
MES-LB2
MES-MC
JES-0
JES-LB
JES-LB2
JES-MC
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of Wall Times

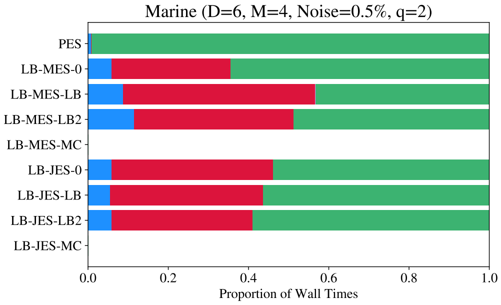

Penicillin (D=7, M=3, Noise=1%, q=1)
PES
MES-0
MES-LB
MES-LB2
MES-MC
JES-0
JES-LB
JES-LB2
JES-MC
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of Wall Times


Marine (D=6, M=4, Noise=0.5%, q=1)
PES
MES-0
MES-LB
MES-LB2
MES-MC
JES-0
JES-LB
JES-LB2
JES-MC
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of Wall Times


ZDT2 (D=6, M=2, Noise=10%, q=2)
PES
LB-MES-0
LB-MES-LB
LB-MES-LB2
LB-MES-MC
LB-JES-0
LB-JES-LB
LB-JES-LB2
LB-JES-MC
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of Wall Times


SnAr (D=4, M=2, Noise=3%, q=2)
PES
LB-MES-0
LB-MES-LB
LB-MES-LB2
LB-MES-MC
LB-JES-0
LB-JES-LB
LB-JES-LB2
LB-JES-MC
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of Wall Times


Penicillin (D=7, M=3, Noise=1%, q=2)
PES
LB-MES-0
LB-MES-LB
LB-MES-LB2
LB-MES-MC
LB-JES-0
LB-JES-LB
LB-JES-LB2
LB-JES-MC
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of Wall Times


Marine (D=6, M=4, Noise=0.5%, q=2)
PES
LB-MES-0
LB-MES-LB
LB-MES-LB2
LB-MES-MC
LB-JES-0
LB-JES-LB
LB-JES-LB2
LB-JES-MC
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of Wall Times

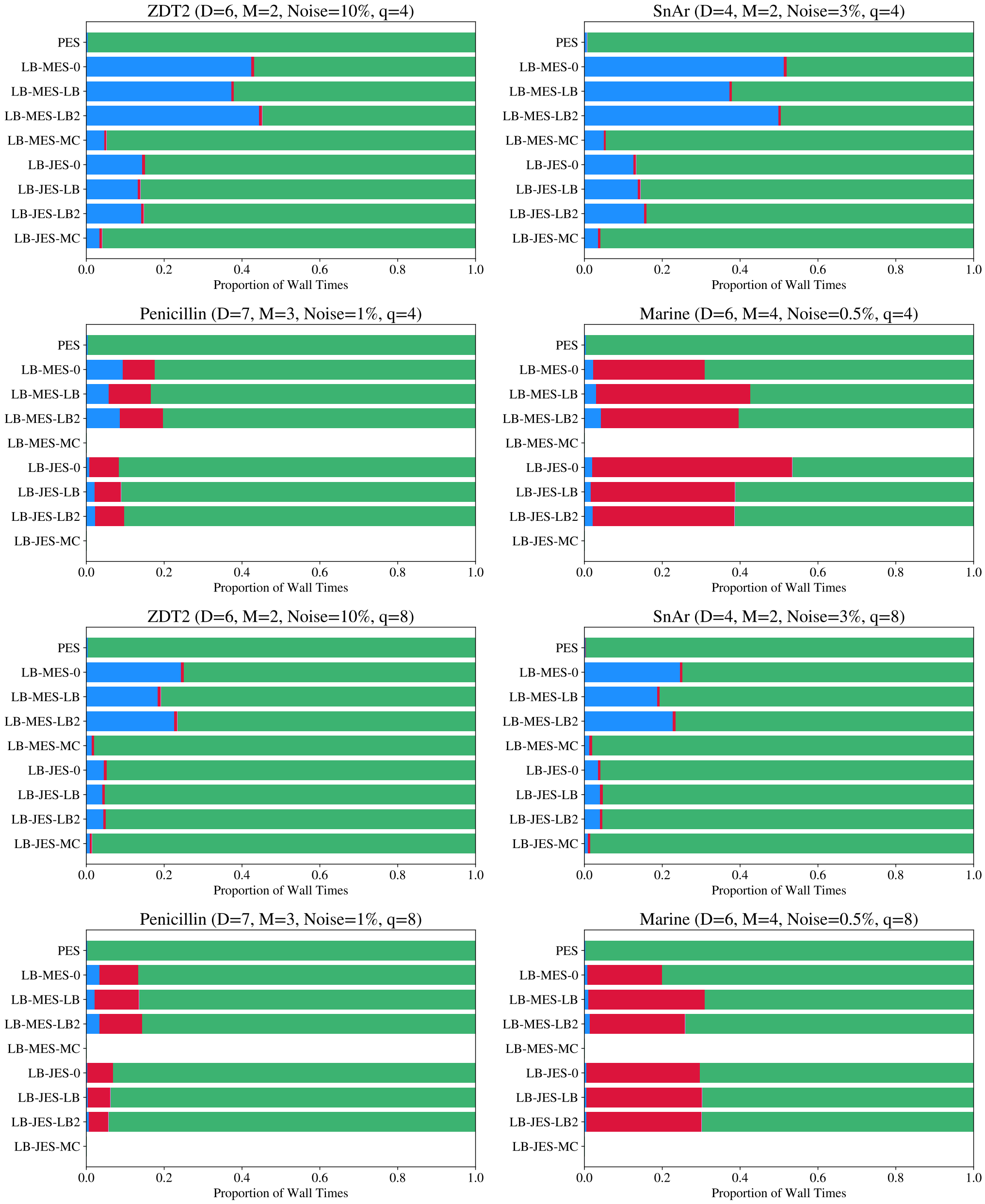


Figure 29: A bar chart comparison of the mean wall times proportions for the information-theoretic acquisition stage. The initialization step includes generation of base samples for the Monte Carlo estimates, the conditioning for JES and the initial expectation propagation steps for PES. The Pareto sampling 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.

M 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, 28, 31, 33, 41, 42, 67, 82]. We propose a simple extension for the JES acquisition function, similar to [82], 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 $k = 1, \dots, K$. 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 [34] 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 [31, 39, 42, 80]. 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 [80], 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 [32, 62, 72] 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, 48, 59, 60, 78, 81, 92, 95]. 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.

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*, 2013. Cited on page 6.

[2] Adelchi Azzalini. A Class of Distributions Which Includes the Normal Ones. *Scandinavian Journal of Statistics*, 1985. Cited on page 22.

[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*. 2020. Cited on pages 8, 33, and 36.

[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*, 2019. Cited on pages 2, 3, 19, 20, and 37.

[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*, 2020. Cited on page 49.

[6] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Output Space Entropy Search Framework for Multi-Objective Bayesian Optimization. *Journal of Artificial Intelligence Research*, 2021. Cited on pages 2 and 49.

[7] Felix Berkenkamp, Andreas Krause, and Angela P. Schoellig. Bayesian optimization with safety Constraints: Safe and automatic parameter tuning in robotics. *Machine Learning*, 2021. Cited on page 1.

[8] Mickael Binois, Victor Picheny, Patrick Taillandier, and Abderrahmane Habbal. The Kalai-Smorodinsky solution for many-objective Bayesian optimization. *Journal of Machine Learning Research*, 2020. Cited on page 2.

[9] Mickael Binois, Abderrahmane Habbal, and Victor Picheny. A game theoretic perspective on Bayesian multi-objective optimization. *arXiv:2104.14456*, 2021. Cited on page 2.

[10] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, 2020. Cited on pages 8 and 33.

[11] Salomon Bochner, Monotonic Functions, Stieltjes Integrals, Harmonic Analysis, Morris Tenenbaum, and Harry Pollard. *Lectures on Fourier Integrals*. Princeton University Press, 1959. Cited on page 18.

[12] Eric Bradford, Artur M. Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *Journal of Global Optimization*, 2018. Cited on pages 8, 18, and 35.

[13] 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*, 2010. Cited on pages 3 and 18.

[14] Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 2016. Cited on page 1.

[15] Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel Gaussian Process Optimization with Upper Confidence Bound and Pure Exploration. In *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, 2013. Cited on page 30.

[16] Ivo Couckuyt, Dirk Deschrijver, and Tom Dhaene. Fast calculation of multiobjective probability of improvement and expected improvement criteria for Pareto optimization. *Journal of Global Optimization*, 2014. Cited on page 4.

[17] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition edition, 2006. Cited on pages 6, 19, and 21.

[18] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization. In *Advances in Neural Information Processing Systems*. 2020. Cited on pages 1, 8, 34, and 35.

[19] 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*, 2021. Cited on pages 1, 8, and 35.

[20] Samuel Daulton, Sait Cakmak, Maximilian Balandat, Michael A. Osborne, Enlu Zhou, and Eytan Bakshy. Robust Multi-Objective Bayesian Optimization Under Input Noise. In *International Conference on Machine Learning*. 2022. Cited on page 36.

[21] Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-Objective Bayesian Optimization over High-Dimensional Search Spaces. In *Uncertainty in Artificial Intelligence*, 2022. Cited on page 1.

[22] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002. Cited on pages 8, 28, and 34.

[23] Jingda Deng and Qingfu Zhang. Approximating Hypervolume and Hypervolume Contributions Using Polar Coordinate. *IEEE Transactions on Evolutionary Computation*, 2019. Cited on pages 8 and 32.

[24] 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, 2019. Cited on page 1.

[25] 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*. 2017. Cited on page 28.

[26] 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*, 2006. Cited on page 1.

[27] Kobi C. Felton, Jan G. Rittig, and Alexei A. Lapkin. Summit: Benchmarking Machine Learning Methods for Reaction Optimisation. *Chemistry–Methods*, 2021. Cited on pages 1 and 36.

[28] 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*, 2021. Cited on page 49.

[29] Peter I. Frazier. A Tutorial on Bayesian Optimization. *arXiv:1807.02811*, 2018. Cited on pages 3 and 18.

[30] 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*. 2018. Cited on pages 8, 28, and 33.

[31] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints. *Neurocomputing*, 2019. Cited on pages 2, 8, 20, 28, 29, 34, 35, 37, and 49.

[32] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing*, 2020. Cited on page 49.

[33] Eduardo C. Garrido-Merchán and Daniel Hernández-Lobato. Parallel Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints. *arXiv:2004.00601*, 2021. Cited on pages 2, 8, 29, 35, and 49.

[34] Michael A. Gelbart, Jasper Snoek, and Ryan P. Adams. Bayesian Optimization with Unknown Constraints. In *Uncertainty in Artificial Intelligence*, 2014. Cited on page 49.

[35] 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*, 2018. Cited on page 1.

[36] Robert B. Gramacy, Annie Sauer, and Nathan Wycoff. Triangulation candidates for Bayesian optimization. *arXiv:2112.07457*, 2021. Cited on page 18.

[37] 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*, 2020. Cited on page 33.

[38] Philipp Hennig and Christian J. Schuler. Entropy Search for Information-Efficient Global Optimization. *Journal of Machine Learning Research*, 2012. Cited on page 3.

[39] Daniel Hernandez-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive Entropy Search for Multi-objective Bayesian Optimization. In *International Conference on Machine Learning*. 2016. Cited on pages 2, 3, 35, and 49.

[40] 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*. 2014. Cited on pages 3, 18, 30, and 35.

[41] Jose Miguel Hernandez-Lobato, Michael Gelbart, Matthew Hoffman, Ryan Adams, and Zoubin Ghahramani. Predictive Entropy Search for Bayesian Optimization with Unknown Constraints. In *International Conference on Machine Learning*. 2015. Cited on page 49.

[42] 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*, 2016. Cited on page 49.

[43] Fred J. Hickernell, Christiane Lemieux, and Art B. Owen. Control Variates for Quasi-Monte Carlo. *Statistical Science*, 2005. Cited on page 26.

[44] Matthew W Hoffman and Zoubin Ghahramani. Output-Space Predictive Entropy Search for Flexible Global Optimization. In *NIPS Workshop on Bayesian Optimization*, 2015. Cited on page 3.

[45] 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*, 2017. Cited on pages 8 and 36.

[46] Carl Hvarfner, Frank Hutter, and Luigi Nardi. Joint Entropy Search For Maximally-Informed Bayesian Optimization. *arXiv:2206.04771*, 2022. Cited on page 2.

[47] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to Specify a Reference Point in Hypervolume Calculation for Fair Performance Comparison. *Evolutionary Computation*, 2018. Cited on page 7.

[48] Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity Bayesian Optimisation with Continuous Approximations. In *International Conference on Machine Learning*. 2017. Cited on page 49.

[49] Tarun Kathuria, Amit Deshpande, and Pushmeet Kohli. Batched Gaussian Process Bandit Optimization via Determinantal Point Processes. In *Advances in Neural Information Processing Systems*. 2016. Cited on page 30.

[50] Andy J. Keane. Statistical Improvement Criteria for Use in Multiobjective Design Optimization. *AIAA Journal*, 2012. Cited on page 4.

[51] Joshua Knowles. ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 2006. Cited on pages 1, 8, and 35.

[52] Mina Konakovic Lukovic, Yunsheng Tian, and Wojciech Matusik. Diversity-Guided Multi-Objective Bayesian Optimization With Batch Evaluations. In *Advances in Neural Information Processing Systems*. 2020. Cited on page 2.

[53] Andreas Krause and Daniel Golovin. Submodular Function Maximization. In *Tractability*, pages 71–104. Cambridge University Press, Cambridge, 2013. Cited on page 27.

[54] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 2012. Cited on pages 27 and 30.

[55] Renaud Lacour, Kathrin Klamroth, and Carlos M. Fonseca. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research*, 2017. Cited on pages 4, 28, and 34.

[56] Qiaohao Liang and Lipeng Lai. Scalable Bayesian Optimization Accelerates Process Optimization of Penicillin Production. In *NeurIPS 2021 AI for Science Workshop*, 2021. Cited on pages 8 and 36.

[57] Xingchao Liu, Xin Tong, and Qiang Liu. Profiling pareto front with multi-objective stein variational gradient descent. In *Advances in Neural Information Processing Systems*. 2021. Cited on page 9.

[58] Thomas P. Minka. Expectation Propagation for Approximate Bayesian Inference. In *Uncertainty in Artificial Intelligence*, 2001. Cited on pages 3 and 28.

[59] Henry B. Moss, David S. Leslie, Javier Gonzalez, and Paul Rayson. GIBBON: General-purpose Information-Based Bayesian Optimisation. *Journal of Machine Learning Research*, 2021. Cited on pages 2, 4, 6, 19, 27, and 49.

[60] 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, 2021. Cited on pages 19 and 49.

[61] Willie Neiswanger, Ke Alexander Wang, and Stefano Ermon. Bayesian Algorithm Execution: Estimating Computable Properties of Black-box Functions Using Mutual Information. In *International Conference on Machine Learning*. 2021. Cited on page 19.

[62] 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*, 2020. Cited on page 49.

[63] 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*, 2020. Cited on page 1.

[64] Biswajit Paria, Kirthevasan Kandasamy, and Barnabás Póczos. A Flexible Framework for Multi-Objective Bayesian Optimization using Random Scalarizations. In *Uncertainty in Artificial Intelligence*. 2020. Cited on page 1.

[65] Michael G. Parsons and Randall L. Scott. Formulation of Multicriterion Design Optimization Problems for Solution With Scalar Numerical Optimization Methods. *Journal of Ship Research*, 2004. Cited on pages 9, 36, and 37.

[66] 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*. 2019. Cited on pages 8, 33, and 34.

[67] Valerio Perrone, Iaroslav Shcherbatyi, Rodolphe Jenatton, Cedric Archambeau, and Matthias Seeger. Constrained Bayesian Optimization with Max-Value Entropy Search. *NeurIPS Workshop on Meta-Learning*, 2019. Cited on page 49.

[68] Victor Picheny. Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, 2015. Cited on pages 2 and 4.

[69] Victor Picheny, Mickael Binois, and Abderrahmane Habbal. A Bayesian optimization approach to find Nash equilibria. *Journal of Global Optimization*, 2019. Cited on page 2.

[70] Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*. 2008. Cited on page 18.

[71] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2006. Cited on pages 3 and 18.

[72] Binxin Ru, Ahsan Alvi, Vu Nguyen, Michael A. Osborne, and Stephen Roberts. Bayesian Optimisation over Multiple Continuous and Categorical Inputs. In *International Conference on Machine Learning*. 2020. Cited on page 49.

[73] Pratyush Sen and Jian-Bo Yang. *Multiple Criteria Decision Support in Engineering Design*. Springer London, 1998. Cited on pages 9, 36, and 37.

[74] Amar Shah and Zoubin Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*. 2015. Cited on page 3.

[75] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 2016. Cited on pages 3 and 18.

[76] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*. 2012. Cited on page 1.

[77] 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*. 2015. Cited on page 1.

[78] Jialin Song, Yuxin Chen, and Yisong Yue. A General Framework for Multi-fidelity Bayesian Optimization with Gaussian Processes. In *International Conference on Artificial Intelligence and Statistics*. 2019. Cited on page 49.

[79] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise Safe Bayesian Optimization with Gaussian Processes. In *International Conference on Machine Learning*. 2018. Cited on page 1.

[80] 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*. 2020. Cited on pages 2, 3, 4, 5, 8, 19, 20, 25, 35, 37, and 49.

[81] 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 *International Conference on Machine Learning*, 37. 2020. Cited on pages 19 and 49.

[82] Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Sequential- and Parallel-Constrained Max-value Entropy Search via Information Lower Bound. *arXiv:2102.09788*, 2021. Cited on page 49.

[83] Ryoji Tanabe and Hisao Ishibuchi. An Easy-to-use Real-world Multi-objective Optimization Problem Suite. *Applied Soft Computing*, 2020. Cited on pages 9, 36, and 37.

[84] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 2008. Cited on page 3.

[85] 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*, 2020. Cited on pages 33 and 34.

[86] Zi Wang and Stefanie Jegelka. Max-value Entropy Search for Efficient Bayesian Optimization. In *International Conference on Machine Learning*. 2017. Cited on pages 3, 4, 18, 20, 35, and 37.

[87] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched High-dimensional Bayesian Optimization via Structural Kernel Learning. In *International Conference on Machine Learning*. 2017. Cited on page 30.

[88] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for Bayesian optimization. In *Advances in Neural Information Processing Systems*. 2018. Cited on pages 7, 26, and 28.

[89] James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently Sampling Functions from Gaussian Process Posteriors. In *International Conference on Machine Learning*. 2020. Cited on pages 18, 19, and 27.

[90] James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Peter Deisenroth. Pathwise Conditioning of Gaussian Processes. *Journal of Machine Learning Research*, 2021. Cited on pages 18 and 19.

[91] 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*, 2019. Cited on page 1.

[92] Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical Multi-fidelity Bayesian Optimization for Hyperparameter Tuning. In *Uncertainty in Artificial Intelligence*. 2020. Cited on page 49.

[93] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Efficient computation of expected hypervolume improvement using box decomposition algorithms. *Journal of Global Optimization*, 2019. Cited on page 1.

[94] Richard Zhang and Daniel Golovin. Random Hypervolume Scalarizations for Provable Multi-Objective Black Box Optimization. In *International Conference on Machine Learning*. 2020. Cited on pages 8 and 32.

[95] Yehong Zhang, Trong Nghia Hoang, Bryan Kian Hsiang Low, and Mohan Kankanhalli. Information-Based Multi-Fidelity Bayesian Optimization. *NIPS Workshop on Bayesian Optimization*, 2017. Cited on page 49.

[96] Eckart Zitzler and Lothar Thiele. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In *Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, 1998. Cited on page 7.

[97] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 2000. Cited on page 36.

[98] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 2003. Cited on pages 1 and 7.

[99] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, 2007. Cited on page 33.