MUMBO: MUlti-task Max-value Bayesian Optimization*

Henry B. Moss[1]✉, David S. Leslie[2], and Paul Rayson[3]

[1] STOR-i Centre for Doctoral Training, Lancaster University
[2] Dept. of Mathematics and Statistics, Lancaster University
[3] School of Computing and Communications, Lancaster University
h.moss@lancaster.ac.uk

Abstract. We propose MUMBO, the first high-performing yet computationally efficient acquisition function for multi-task Bayesian optimization. Here, the challenge is to perform efficient optimization by evaluating low-cost functions somehow related to our true target function. This is a broad class of problems including the popular task of multi-fidelity optimization. However, while information-theoretic acquisition functions are known to provide state-of-the-art Bayesian optimization, existing implementations for multi-task scenarios have prohibitive computational requirements. Previous acquisition functions have therefore been suitable only for problems with both low-dimensional parameter spaces and function query costs sufficiently large to overshadow very significant optimization overheads. In this work, we derive a novel multi-task version of entropy search, delivering robust performance with low computational overheads across classic optimization challenges and multi-task hyperparameter tuning. MUMBO is scalable and efficient, allowing multi-task Bayesian optimization to be deployed in problems with rich parameter and fidelity spaces.

Keywords: Bayesian Optimization · Gaussian Processes.

1 Introduction

The need to efficiently optimize functions is ubiquitous across machine learning, operational research and computer science. Many such problems have special structures that can be exploited for efficient optimization, for example gradient-based methods on cheap-to-evaluate convex functions, and mathematical programming for heavily constrained problems. However, many optimization problems do not have such clear properties.

Bayesian optimization (BO) is a general method to efficiently optimize 'black-box' functions for which we have weak prior knowledge, typically characterized by expensive and noisy function evaluations, a lack of gradient information, and high levels of non-convexity (see [1] for a comprehensive review). By

* Supported by EPSRC and the STOR-i Centre for Doctoral Training.

sequentially deciding where to make each evaluation as the optimization progresses, BO is able to direct resources into promising areas and so efficiently explore the search space. In particular, a highly effective and intuitive search is achieved through **information-theoretic** BO, where we seek to sequentially reduce our uncertainty (measured in terms of differential entropy) in the location of the optima with each successive function evaluation [2,3].

For optimization problems where we can evaluate low-cost functions somehow related to our true objective function, **Multi-task** (MT) BO (as first introduced by [4]) provides additional efficiency gains. A popular subclass of MT BO problems is **multi-fidelity** (MF) BO, where the set of related functions can be meaningfully ordered by their similarity to the objective function. Unfortunately, performing BO over MT spaces has previously required complicated approximation schemes that scale poorly with dimension [4,5], limiting the applicability of information-theoretic arguments to problems with both low-dimensional parameter spaces and function query costs sufficiently large to overshadow very significant optimization overheads. Therefore, MT BO has so far been restricted to considering simple structures at a large computational cost. Despite this restriction, MT optimization has wide-spread use across physical experiments [6,7,8], environmental modeling [9], and operational research [10,11,12].

For expositional simplicity, this article focuses primarily on examples inspired by tuning the hyper-parameters of machine learning models. Such problems have large environmental impact [13], requiring multiple days of computation to collect even a single (often highly noisy) performance estimate. Consequently, these problems have been proven a popular and empirically successful application of BO [14]. MF applications for hyper-parameter tuning dynamically control the reliability (in terms of bias and noise) of each hyper-parameter evaluation [15,16,17,18,19] and can reduce the computational cost of tuning complicated models by orders of magnitude over standard BO. Orthogonal savings arise from considering hyper-parameter tuning in another MT framework; FASTCV [4] recasts tuning by K-fold cross-validation (CV) [20] into the task of simultaneously optimizing the K different evaluations making a single K-fold CV estimate.

Information-theoretic arguments are particularly well suited to such MT problems as they provide a clear measure of the utility (the information gained) of making an evaluation on a particular subtask. This utility then can be balanced with computational cost, providing a single principled decision [4,17,21,5]. Despite MT BO being a large sub-field in its own right, there exist only a few alternatives to information-theoretic acquisition functions. Alternative search strategies include extensions of standard BO acquisition functions, including knowledge gradient (KG) [22,23], expected improvement (EI) [4,24,16], and upper-confidence bound (UCB) [18,19]. KG achieves efficient optimization but incurs a high computational overhead. The MT extensions of EI and UCB, although computationally cheap, lack a clear notion of utility and consequently rely on two-stage heuristics, where a hyper-parameter followed by a task are chosen as two separate decisions. Moreover, unlike our proposed work, the performance of MT variants of UCB and EI depends sensitively on problem-specific param-

eters which require careful tuning, often leading to poor performance in practical tasks. Information-theoretic arguments have produced the MF BO hyper-parameter tuner FABOLAS [17], out-competing approaches across richer fidelity spaces based on less-principled acquisitions [19]. This success motivates our work to provide scalable entropy reduction over MT structures.

We propose **MUMBO**, a novel, scalable and computationally light implementation of information-theoretic BO for general MT frameworks. Inspired by the work of [25], we seek reductions in our uncertainty in the value of the objective function at its optima (a single-dimensional quantity) rather than our uncertainty in the location of the optima itself (a random variable with the same dimension as our search space). MUMBO enjoys three major advantages over current information-theoretic MT approaches:

- MUMBO has a simple and scalable formulation requiring routine one-dimensional approximate integration, irrespective of the search space dimensions,
- MUMBO is designed for general MT and MF BO problems across both continuous and discrete fidelity spaces,
- MUMBO outperforms current information-theoretic MT BO with a significantly reduced computational cost.

Parallel work [26] presents essentially the same acquisition function but restricted to discrete multi-fidelity problems from the material sciences. Our article provides a different derivation and general presentation of the method which enables deployment with both discrete and continuous fidelity spaces in general MT BO (including MF). We also provide an implementation in a major BO toolbox and examples across synthetic and hyper-parameter tuning tasks.

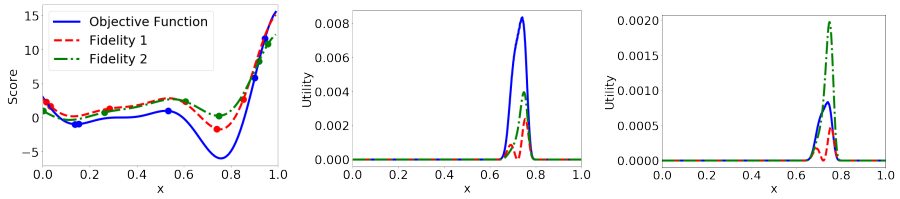
2 Problem Statement and Background

We now formalize the goal of MT BO, introducing the notation used throughout this work. The goal of BO is to find the maximzer

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\, g(\mathbf{x}) \tag{1}$$

of a function g over a d-dimensional set of feasible choices $\mathcal{X} \subset \mathbb{R}^d$ spending as little computation on function evaluations as possible.

Standard BO seeks to solve (1) by sequentially collecting noisy observations of g. By fitting a Gaussian process (GP) [27], a non-parametric model providing regression over all functions of a given smoothness (to be controlled by a choice of kernel k), we are able to quantify our current belief about which areas of the search space maximize our objective function. An acquisition function $\alpha_n(\mathbf{x}) : \mathcal{X} \to \mathbb{R}$ uses this belief to predict the utility of making any given evaluation, producing large values at 'reasonable' locations. A standard acquisition function [2] is the expected amount of information provided by each evaluation about the location of the maximum. Therefore after making n evaluations, BO will automatically next evaluate $\mathbf{x}_{n+1} = \operatorname{argmax}_{\mathbf{x}\in\mathcal{X}} \alpha_n(\mathbf{x})$.



(a) Collected Observations (b) Information gain (c) Gain per unit cost

Fig. 1: Seeking the minimum of the 1D Forrester function (blue) with access to two low-fidelity approximations at $\frac{1}{2}$ (red) and $\frac{1}{5}$ (green) the cost of querying the true objective. Although we learn the most from directly querying the objective function, we can learn more per unit cost by querying the roughest fidelity.

2.1 Multi-task Bayesian Optimization

Suppose that instead of querying g directly, we can alternatively query a (possibly infinite) collection of related functions indexed by $\mathbf{z} \in \mathcal{Z}$ (henceforth referred to as our fidelity space). We then collect the (noisy) observations $D_n = \{(\mathbf{x}_t, \mathbf{z}_t, y_t)\}$ for $y_t = f(\mathbf{x}_t, \mathbf{z}_t) + \epsilon_t$, where $f(\mathbf{x}, \mathbf{z})$ is the result of querying parameter $\mathbf{x}$ on fidelity $\mathbf{z}$, and ϵ_t is Gaussian noise. If these alternative functions f are cheaper to evaluate and we can learn their relationship with g, then we have access to cheap sources of information that can be used to help find the maximizer of the true task of interest.

2.2 Multi-task acquisition functions

The key difference between standard BO and MT BO is that our acquisition function must be able to not only choose the next location, but also which fidelity to evaluate, balancing computational cost with how much we expect to learn about the maximum of g. Therefore, we require an extended acquisition function $\alpha_n : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ and a cost function $c : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}^+$, measuring the utility and cost of evaluating location $\mathbf{x}$ at fidelity $\mathbf{z}$ (as demonstrated in Figure 1c). In Section 4, we consider problems both where this cost function is known *a priori* and where it is unknown but estimated using an extra GP [14]. In this work, we seek to make the evaluation that provides the largest information gain per unit cost, i.e. maximizing the ratio

$$(\mathbf{x}_{n+1}, \mathbf{z}_{n+1}) = \underset{(\mathbf{x},\mathbf{z}) \in \mathcal{X} \times \mathcal{Z}}{\operatorname{argmax}} \frac{\alpha_n(\mathbf{x}, \mathbf{z})}{c(\mathbf{x}, \mathbf{z})}. \tag{2}$$

2.3 Multi-task models

To perform MT BO, our underlying Gaussian process model must be extended across the fidelity space. By defining a kernel over $\mathcal{X} \times \mathcal{Z}$, we can learn predictive distributions after n observations with means $\mu^n(\mathbf{x}, \mathbf{z})$ and co-variances

$\Sigma^n((\mathbf{x}, \mathbf{z}), (\mathbf{x}', \mathbf{z}'))$ from which $\alpha_n(\mathbf{x}, \mathbf{z})$ can be calculated. Although increasing the dimension of the kernel for $\mathcal{X}$ to incorporate $\mathcal{Z}$ provides a very flexible model, it is argued by [19] that overly flexible models can harm optimization speed by requiring too much learning, restricting the sharing of information across the fidelity space. Therefore, it is common to use more restrictive separable kernels that better model specific aspects of the given problem.

A common kernel for discrete fidelity spaces is the intrinsic coregionalization kernel of [28] (as used in Figure 1). This kernel defines a co-variance between hyper-parameter and fidelity pairs of

$$k((\mathbf{x}, z), (\mathbf{x}', z')) = k_{\mathcal{X}}(\mathbf{x}, \mathbf{x}') \times B(z, z'), \tag{3}$$

for a base kernel $k_{\mathcal{X}}$ and a positive semi-definite $|\mathcal{Z}| \times |\mathcal{Z}|$ matrix B (set by maximizing the model likelihood alongside the other kernel parameters). B represents the correlation between different fidelities, allowing the sharing of information across the fidelity space. See Section 4 for additional standard MF kernels.

2.4 Information-theoretic MT BO

Existing methods for information-theoretic MT BO seek to maximally reduce our uncertainty in the location of the maximizer $\mathbf{x}^* = \text{argmax}_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x})$. Following the work of [2], uncertainty in the value of $\mathbf{x}^*$ is measured as its differential entropy $H(\mathbf{x}^*) = -\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}^*}} (\log p_{\mathbf{x}^*}(\mathbf{x}))$, where $p_{\mathbf{x}^*}$ is the probability density function of $\mathbf{x}^*$ according to our current GP model. For MT optimization, we require knowledge of the amount of information provided about the location of $\mathbf{x}^*$ from making an evaluation at $\mathbf{x}$ on fidelity $\mathbf{z}$, measured as the mutual information

$$I(y(\mathbf{x}, \mathbf{z}); \mathbf{x}^* | D_n) = H(\mathbf{x}^* | D_n) - \mathbb{E}_y \left[H(\mathbf{x}^* | y(\mathbf{x}, \mathbf{z}), D_n) \right]$$

between an evaluation $y(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}, \mathbf{z}) + \epsilon$ and $\mathbf{x}^*$, where the expectation is over $p(y(\mathbf{x}, \mathbf{z}) | D_n)$ (see [29] for an introduction to information theory).

Successively evaluating the parameter-fidelity pair that provides the largest information gain per unit of evaluation cost provides the entropy search acquisition function used by [4] and [17], henceforth referred to as the MTBO acquisition function. Unfortunately, the calculation of MTBO relies on sampling-based approximations to the non-analytic distribution of $\mathbf{x}^* \,|\, D_n$. Such approximations scale poorly in both cost and performance with the dimensions of our search space (as demonstrated in Section 4). A modest computational saving can be made for standard BO problems by exploiting the symmetric property of mutual information, producing the predictive entropy search (PES) of [3]. However, PES still requires approximations of $\mathbf{x}^* \,|\, D_n$ and it is unclear how to extend this approach across MT frameworks.

3 MUMBO

In this work, we extend the computationally efficient information-theoretic acquisition function of [25] to MT BO. With their max-value entropy-search acquisi-

tion function (MES), they demonstrate that seeking to reduce our uncertainty in the value of $g^* = g(\mathbf{x}^*)$ provides an equally effective search strategy as directly minimizing the uncertainty in the location $\mathbf{x}^*$, but with significantly reduced computation. Similarly, MUMBO seeks to compute the information gain

$$\alpha_n^{MUMBO}(\mathbf{x}, \mathbf{z}) = H(y(\mathbf{x}, \mathbf{z}) \mid D_n) - \mathbb{E}_{g^*}[H(y(\mathbf{x}, \mathbf{z}) \mid g^*, D_n)], \tag{4}$$

which can then be combined with the evaluation cost $c(\mathbf{x}, \mathbf{z})$ (following (2)). Here the expectation is over our current uncertainty in the value of $g^*|D_n$.

3.1 Calculation of MUMBO

Although extending MES to MT scenarios retains the intuitive formulation and the subsequent principled decision-making of the original MES, we require a novel non-trivial calculation method to maintain its computational efficiency for MT BO. We now propose a strategy for calculating the MUMBO acquisition function that requires the approximation of only single-dimensional integrals irrespective of the dimensions of our search space.

The calculation of our MUMBO acquisition function (4) for arbitrary $\mathbf{x}$ and $\mathbf{z}$ must be efficient as each iteration of BO requires a full maximization of (4) over $\mathbf{x}$ and $\mathbf{z}$ (i.e 2). For ease of notation we drop the dependence on $\mathbf{x}$ and $\mathbf{z}$, so that g denotes the target function value at $\mathbf{x}$, f denotes the evaluation of $\mathbf{x}$ at fidelity $\mathbf{z}$, and y denotes the (noisy) observed value of $f(\mathbf{x}, \mathbf{z})$. Since BO fits a Gaussian process to the underlying functions, our assumptions about g and y imply that their joint predictive distribution is a bivariate Gaussian; with expectation, variance and correlation derived from our GP (as shown in Appendix A) and denoted by (μ_g, μ_f), $(\sigma_g^2, \sigma_f^2 + \sigma^2)$ and ρ respectively. These values summarize our current uncertainty in g and f and how useful making an evaluation y will be for learning about g. Note that access to this simple two-dimensional predictive distribution is all that is needed to calculate MUMBO (4).

The first term of (4) is the differential entropy of a Gaussian distribution and so can be calculated analytically as $\frac{1}{2}\log(2\pi e(\sigma_f^2 + \sigma^2))$. The second term of (4) is an expectation over the maximum value of the true objective g^*, which can be approximated using a Monte Carlo approach; we use [25]'s method to approximately sample a set of N samples $G = \{g_1, \ldots, g_N\}$ from $g^* \mid D_n$, using a mean-field approximation and extreme value theory.

It remains to calculate the quantity inside the expectation for a given value of g^*. The equivalent quantity in the original MES (without fidelity considerations) was analytically tractable, but we show that for MUMBO this term is intractable. In particular, we show that $y \mid g < g^*$ follows an extended-skew Gaussian (ESG) distribution [30,31] in Appendix A. Unfortunately, [32] have shown that there is no analytical form for the differential entropy of an ESG. Therefore, after manipulations presented also in Appendix A and reintroducing dependence on $\mathbf{x}$ and $\mathbf{z}$, we re-express (4) as

$$\alpha_n^{MUMBO}(\mathbf{x},\mathbf{z}) = \frac{1}{N}\sum_{g^*\in G}\left[\rho(\mathbf{x},\mathbf{z})^2\frac{\gamma_{g^*}(\mathbf{x})\phi(\gamma_{g^*}(\mathbf{x}))}{2\Phi(\gamma_{g^*}(\mathbf{x}))} - \log(\Phi(\gamma_{g^*}(\mathbf{x}))) + \mathbb{E}_{\theta\sim Z_{g^*}(\mathbf{x},\mathbf{z})}\left[\log\left(\Phi\left\{\frac{\gamma_{g^*}(\mathbf{x})-\rho(\mathbf{x},\mathbf{z})\theta}{\sqrt{1-\rho^2(\mathbf{x},\mathbf{z})}}\right\}\right)\right]\right], \quad (5)$$

where Φ and ϕ are the standard normal cumulative distribution and probability density functions, $\gamma_{g^*}(\mathbf{x}) = \frac{g^*-\mu_g(\mathbf{x})}{\sigma_g(\mathbf{x})}$ and $Z_{g^*}(\mathbf{x},\mathbf{z})$ is an ESG (with probability density function provided in Appendix A).

Expression (5) is analytical except for the final term, which must be approximated for each of the N samples of g^* making up the Monte Carlo estimate. Crucially, this is just a single-dimensional integral of an analytic expression and, hence, can be quickly and accurately approximated using standard numerical integration techniques. We present MUMBO within a BO loop as Algorithm 1.

Algorithm 1 MUlti-fidelity and MUlti-task Max-value Bayesian Optimization: MUMBO

function MUMBO(budget B, N samples of g^*)
  Initialize $n \leftarrow 0$, $b \leftarrow 0$
  Collect initial design D_0
  while $b < B$ **do**
    Begin new iteration $n \leftarrow n+1$
    Fit GP to the collected observations D_{n-1}
    Simulate N samples of $g^*|D_{n-1}$
    Prepare $\alpha_{n-1}^{MUMBO}(\mathbf{x},\mathbf{z})$ as given by Eq. (5)
    Find the next point and fidelity to query $(\mathbf{x}_n,\mathbf{z}_n) \leftarrow \text{argmax}_{(\mathbf{x},\mathbf{z})} \frac{\alpha_{n-1}^{MUMBO}(\mathbf{x},\mathbf{z})}{c(\mathbf{x},\mathbf{z})}$
    Collect the new evaluation $y_n \leftarrow f(\mathbf{x}_n,\mathbf{z}_n)+\epsilon_n$, $\epsilon_n \sim N(0,\sigma^2)$
    Append new evaluation to observation set $D_n \leftarrow D_{n-1}\bigcup\{(\mathbf{x}_n,\mathbf{z}_n),y_n\}$
    Update spent budget $b \leftarrow b + c(\mathbf{x}_n,\mathbf{z}_n)$
  return Believed optimum across $\{\mathbf{x}_1,..,\mathbf{x}_n\}$

3.2 Interpretation of MUMBO

We provide intuition for (5) by relating MUMBO to an established BO acquisition function. In the formulation of MUMBO (5), we see that for a fixed parameter choice $\mathbf{x}$ (and ignoring evaluation costs) this acquisition is maximized by choosing the fidelity z that provides the largest $|\rho(\mathbf{x},\mathbf{z})|$, meaning that the stronger the correlation (either negatively or positively) the more we can learn about the true objective. In fact, if we find a fidelity $\mathbf{z}^*$ that provides evaluations that agree completely with g, then we would have $\rho(\mathbf{x},\mathbf{z}^*) = 1$ and (5) would

collapse to

$$\alpha_n(\mathbf{x}, \mathbf{z}^*) = \frac{1}{N} \sum_{g^* \in G} \left[\frac{\gamma_{g^*}(\mathbf{x})\phi(\gamma_{g^*}(\mathbf{x}))}{2\Phi(\gamma_{g^*}(\mathbf{x}))} - \log(\Phi(\gamma_{g^*}(\mathbf{x}))) \right].$$

This is exactly the same expression presented by [25] in the original implementation of MES, appropriate for standard BO problems where we can only query the function we wish to optimize.

3.3 Computational Cost of MUMBO

The computational complexity of any BO routine is hard to measure exactly, due to the acquisition maximization (2) required before each function query. However, the main contributor to computational costs are the resources required for each calculation of the acquisition function with respect to problem dimension d and the N samples of g^*. Each prediction from our GP model costs $O(d)$, and single-dimensional numerical integration over a fixed grid is $O(1)$. Therefore, a single evaluation of MUMBO can be regarded as an $O(Nd)$ operation. Moreover, as MUMBO relies on the approximation of a single-dimensional integral, we do not require an increase in N to maintain performance as the problem dimension d increases (as demonstrated in Section 4) and so MUMBO scales linearly with problem dimension. In contrast, the MT BO acquisition used by [4] and [17] for information-theoretic MT BO relies on sampling-based approximations of d-dimensional distributions, therefore requiring exponentially increasing sample sizes to maintain performance as dimension increases, rendering them unsuitable for even moderately-sized BO problems. In addition, we note that these current approaches require expensive sub-routines and the calculation of derivative information, making their computational cost for even small d much larger than that of MUMBO.

4 Experiments

We now demonstrate the performance of MUMBO across a range of MT scenarios, showing that MUMBO provides superior optimization to all existing approaches, with a significantly reduced computational overhead compared to current state-of-the-art. As is common in the optimization literature, we first consider synthetic benchmark functions in a discrete MF setting. Next, we extend the challenging continuous MF hyper-parameter tuning framework of FABOLAS and use MUMBO to provide a novel information-theoretic implementation of the MT hyper-parameter tuning framework FASTCV, demonstrating that the performance of this simple MT model can be improved using our proposed fully-principled acquisition function. Finally, we use additional synthetic benchmarks to compare MUMBO against a wider range of existing MT BO acquisition functions.

Alongside the theoretical arguments of this paper, we also provide a software contribution to the BO community of a flexible implementation of MUMBO with

support for Emukit [35]. We use a DIRECT optimizer [36] for the acquisition maximization at each BO step and calculate the single-dimensional integral in our acquisition (5) using Simpson's rule over appropriate ranges (from the known expressions of an ESG's mean and variance derived in Appendix A.1).

4.1 General Experimental Details

Overall, the purpose of our experiments is to demonstrate how MUMBO compares to other acquisition functions when plugged into a set of existing MT problems, focusing on providing a direct comparison with the existing state-of-the-art in information-theoretic MT BO used by [4] and [17] (which we name MTBO). Our main experiments also include the performance of popular low-cost MT acquisition functions MF-GP-UCB [18] and MT expected improvement [4]. In Section 4.5 we expand our comparison to include a wider range of existing BO routines, chosen to reflect popularity and code availability. We include the MF knowledge gradient (MISO-KG)[22][4], an acquisition function with significantly larger computational overheads than MUMBO (and MTBO), as-well as the low-cost acquisition functions of BOCA [19] and MF-SKO [10]. Due to a lack of provided code, and the complexity of their proposed implementations, we were unable to implement multi-fidelity extensions of PES [21,5] or the variant of knowledge-gradient for continuous fidelity spaces [23]. As both PES and knowledge gradient require approximations of quantities with dimensionality equal to the search space, their MT extensions will suffer the same scalability issue as MTBO (and MISO-KG). Finally, to demonstrate the benefit of considering MT frameworks, we also present the standard BO approaches of expected improvement (EI) and max-value entropy search (MES) which query only the true objective.

To test the robustness of the information-theoretic acquisitions we vary the number of Monte Carlo samples N used for both MUMBO and MTBO (denoted as MUMBO-N and MTBO-N). We report both the time taken to choose the next location to query (referred to as the optimization overhead) and the performance of the believed objective function optimizer (the incumbent) as the optimization progresses. For our synthetic examples, we measure performance after n evaluations as the simple regret $R_n = g(\mathbf{x}^*) - g(\hat{\mathbf{x}}_n)$, representing the sub-optimality of the current incumbent $\hat{\mathbf{x}}_n$. Experiments reporting wall-clock timings were performed on single core Intel Xeon 2.30GHz processors. Detailed implementation details are provided in Appendix B.

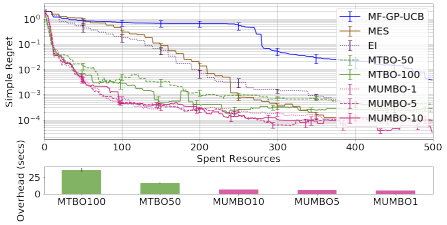
4.2 Discrete Multi-fidelity BO

First, we consider the optimization of synthetic problems, using the intrinsic coregionalization kernel introduced earlier (3). Figure 2 demonstrates the superior performance and light computational overhead of MUMBO across these test

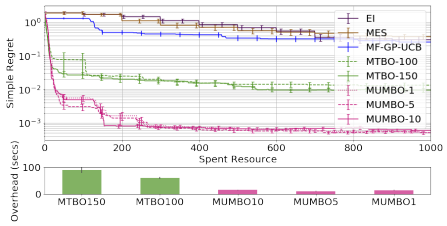
[4] As implemented by the original authors at *https://github.com/misokg/NIPS2017*

functions when we have access to continuous or discrete collections of cheap low-fidelity functions at lower query costs. Although MTBO and MUMBO initially provide comparable levels of optimization, MUMBO quickly provides optimization with substantially higher precision than MTBO and MF-GP-UCB. We delve deeper into the low performance of MF-GP-UCB in Appendix B.1. In addition, MUMBO is able to provide high-precision optimization even when based on a single sample of g^*, whereas MTBO requires 50 samples for reasonable performance on the 2D optimization task, struggles on the 6D task even when based on 200 samples (requiring 20 times the overhead cost of MUMBO), and proved computationally infeasible to provide reasonable 8D optimization (and is therefore not included in Figure 2d).

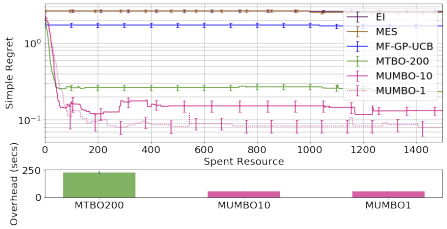
Note that MUMBO based on a single sample of g^* is a more aggressive optimizer, as we only consider a single (highly-likely) max-value. Although less robust than MUMBO-10 on average across our examples, this aggressive behavior can allow faster optimization, but only for certain problems (Figure 2(c)).



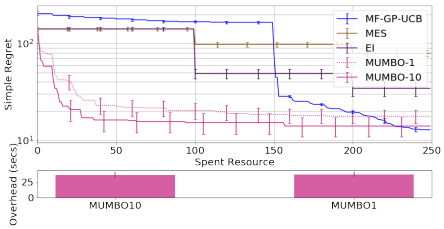
(a) Maximization of the 2D Currin function (2 fidelity levels).



(b) Minimization of 3D Hartmann function (3 fidelity levels).



(c) Minimization of 6D Hartmann function (4 fidelity levels).



(d) Maximization of the 8D Borehole function (2 fidelity levels).

Fig. 2: MUMBO provides high-precision optimization with low computational overheads for discrete MF optimization. We show the means and standard errors across 20 random initializations.

4.3 Continuous Multi-fidelity BO: FABOLAS

FABOLAS [17] is a MF framework for tuning the hyper-parameter of machine learning models whilst dynamically controlling the proportion of available data $z \in (0, 1]$ used for each hyper-parameter evaluation. By using the MTBO acquisition and imposing strong assumptions on the structure of the fidelity space, FABOLAS is able to achieve highly efficient hyper-parameter tuning. The use of a 'degenerate' kernel [27] with basis function $\phi(z) = (z, (1-z)^2)^T$ (i.e performing Bayesian linear regression over this basis) enforces monotonicity and strong smoothness across the fidelity space, acknowledging that when using more computational resources, we expect less biased and less noisy estimates of model performance. These assumptions induce a product kernel over the whole space $\mathcal{X} \times \mathcal{Z}$ of:

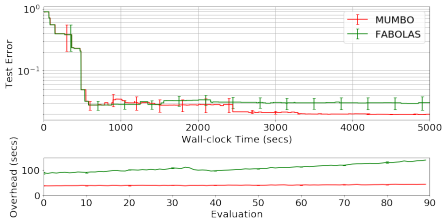
$$k((\mathbf{x}, z), (\mathbf{x}', z')) = k_{\mathcal{X}}(\mathbf{x}, \mathbf{x}')(\phi(z)^T \Sigma_1 \phi(z')),$$

where Σ_1 is a matrix in $\mathbb{R}^{2\times 2}$ to be estimated alongside the parameters of $k_{\mathcal{X}}$. Similarly, evaluation costs are also modeled in log space, with a GP over the basis $\phi_c(z) = (1, z)^T$ providing polynomial computational complexity of arbitrary degree. We follow the original FABOLAS implementation exactly, using MCMC to marginalize kernel parameters over hyper-priors specifically chosen to speed up and stabilize the optimization.

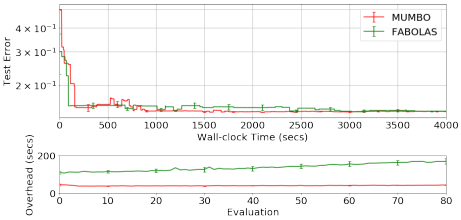
In Figure 3 we replace the MTBO acquisition used within FABOLAS with a MUMBO acquisition, demonstrating improved optimization on two examples from [17]. As the goal of MF hyper-parameter tuning is to find high-performing hyper-parameter configurations after using as few computational resources as possible, including both the fitting of models and calculating the next hyper-parameter and fidelity to query, we present incumbent test error (calculated offline after the full optimization) against wall-clock time. Note that the entire time span considered for our MNIST example is still less than required to try just four hyper-parameter evaluations on the whole data and so we cannot include standard BO approaches in these figures. MUMBO's significantly reduced computational overhead allows twice as many hyper-parameter evaluations as MTBO for the same wall clock time, even though MUMBO consistently queries larger proportions of the data (on average 30% rather than 20% by MTBO). Moreover, unlike MTBO, with an overhead that increases as the optimization progresses, MUMBO remains computationally light-weight throughout and has substantially less variability in the performance of the chosen hyper-parameter configuration. While we do not compare FABOLAS against other hyper-parameter tuning methods, we have demonstrated that, for this well-respected tuner and complicated MF BO problem, that MUMBO provides an improvement in efficiency and a substantial reduction in computational cost.

4.4 Multi-task BO: FASTCV

We now consider the MT framework of FASTCV [4]. Here, we seek the simultaneous optimization of the K performance estimates making up K-fold CV.



(a) Tuning C and gamma for an SVM to minimize MNIST digit classification error.



(b) Tuning C and gamma for an SVM to minimize Vehicle Registration classification error.

Fig. 3: MUMBO provides MF hyper-parameter tuning with a much lower overhead than FABOLAS. We show the means and standard errors based on 5 runs.

Therefore, our objective function g is the average score across a categorical fidelity space $\mathcal{Z} = \{1, .., K\}$. Each hyper-parameter is evaluated on a single fold, with the corresponding evaluations on the remaining folds inferred using the learned between-fold relationship. Therefore, we can evaluate K times as many distinct hyper-parameter choices as when tuning with full K-fold CV whilst retaining the precise performance estimates required for reliable tuning [37,38].

Unlike our other examples, this is not a MF BO problem as our fidelities have the same query cost (at $1/K^{th}$ the cost of the true objective). Recall that all we require to use MUMBO is the predictive joint (bi-variate Gaussian) distribution between an objective function $g(\mathbf{x})$ and fidelity evaluations $f(\mathbf{x}, z)$ for each choice of $\mathbf{x}$. For FASTCV, g corresponds with the average score across folds and so (following our earlier notation) our underlying GP provides;

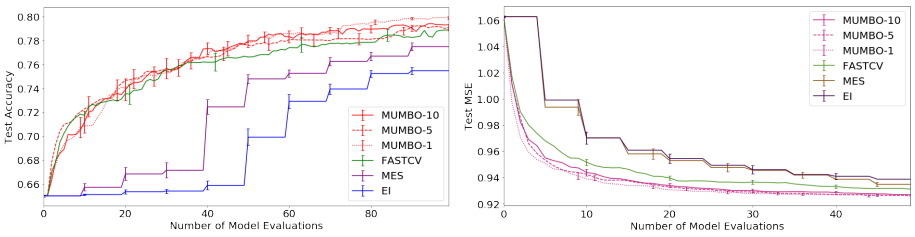
$$\mu_g(\mathbf{x}) = \frac{1}{K}\sum_{z\in\mathcal{Z}} \mu^n(\mathbf{x}, z), \quad \sigma_g(\mathbf{x})^2 = \frac{1}{K^2}\sum_{z\in\mathcal{Z}}\sum_{z'\in\mathcal{Z}} \Sigma^n((\mathbf{x}, z), (\mathbf{x}, z')),$$

where $\mu^n(\mathbf{x}, z)$ is the predictive mean performance of $\mathbf{x}$ on fold z and $\Sigma^n((\mathbf{x}, z), (\mathbf{x}, z'))$ is the predictive co-variance between the evaluations of $\mathbf{x}$ on folds z and z' after n hyper-parameter queries. Similarly, we have the correlation between evaluations of $\mathbf{x}$ on fold z with the average score g as

$$\rho(\mathbf{x}, z) = \frac{\frac{1}{K}\sum_{z'\in\mathcal{Z}} \Sigma^n((\mathbf{x}, z), (\mathbf{x}, z'))}{\sqrt{\sigma_g^2(\mathbf{x})\Sigma^n((\mathbf{x}, z), (\mathbf{x}, z))}},$$

providing all the quantities required to use MUMBO.

In the original implementation of FASTCV, successive hyper-parameter evaluations are chosen using a two-step heuristic based on expected improvement. Firstly they choose the next hyper-parameter $\mathbf{x}$ by maximizing the expected improvement of the predicted average performance and secondly choosing the fold that has the largest fold-specific expected improvement at this chosen hyper-parameter. We instead propose using MUMBO to provide a principled information-theoretic extension to FASTCV. Figure 4 demonstrates that MUMBO provides



(a) Tuning two SVM hyper-parameters to maximize sentiment classification accuracy for IMDB movie reviews by 10-fold CV.

(b) Tuning four hyper-parameters for probabilistic matrix factorization to minimize mean reconstruction error for movie recommendations using 5-fold CV.

Fig. 4: MUMBO provides faster hyper-parameter tuning than the MT framework of FASTCV. We show the mean and standard errors across 40 runs. To measure total computational cost we count each evaluation by K-fold CV as k model fits. Experimental details are included in Appendix B.3.

an efficiency gain over FASTCV, while finding high-performing hyper-parameters substantially faster than standard BO tuning by K-fold CV (where we require K model evaluations for each unique hyper-parameter query.

4.5 Wider Comparison With Existing Methods

Finally, we make additional comparisons with existing MT acquisition functions in Figures 5 and 6. Knowledge-gradient search strategies are designed to provide particularly efficient optimization for noisy functions, however this high performance comes with significant computational overheads. Although providing reasonable early performance on a synthetic noisy MF optimization task (Figure 5), we see that MUMBO is able to provide higher-precision optimization and that, even for this simple 2-d search space, MISO-KG's optimization overheads are magnitudes larger than MUMBO (and MTBO). Figure 6 shows that MUMBO substantially outperforms existing approaches on a continuous MF benchmark. MF-SKO, MF-UCB and BOCA's search strategies are guided by estimating g^* (rather than $\mathbf{x}^*$) and so have comparable computational cost to MUMBO, however, only MUMBO is able to provide high-precision optimization with this low-computational overhead.

5 Conclusions

We have derived a novel computationally light information-theoretic approach for general discrete and continuous multi-task Bayesian optimization, along with an open and accessible code base that will enable users to deploy these methods and improve replicability.

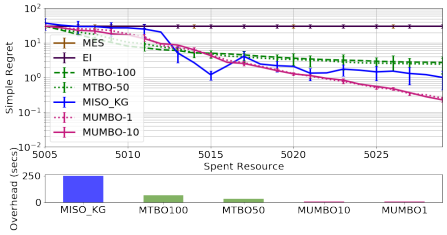


Fig. 5: The 2D noisy Rosenbrock function (2 fidelities).

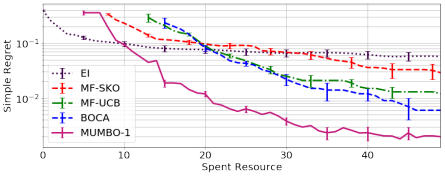


Fig. 6: The 2-d Currin function (1-d continuous fidelity space)

MUMBO reduces uncertainty in the optimal value of the objective function with each subsequent query, and provides principled decision-making across general multi-task structures at a cost which scales only linearly with the dimension of the search space. Consequently, MUMBO substantially outperforms current acquisitions across a range of optimization and hyper-parameter tuning tasks.

References

1. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE (2016)
2. Hennig, P., Schuler, C.J.: Entropy search for information-efficient global optimization. Journal of Machine Learning Research (2012)
3. Hernández-Lobato, J.M., Hoffman, M.W., Ghahramani, Z.: Predictive entropy search for efficient global optimization of black-box functions. In: Neural Information Processing Systems (2014)
4. Swersky, K., Snoek, J., Adams, R.P.: Multi-task Bayesian optimization. In: Neural Information Processing Systems (2013)
5. Zhang, Y., Hoang, T.N., Low, B.K.H., Kankanhalli, M.: Information-based multi-fidelity Bayesian optimization. In: Neural Information Processing Systems: Workshop on Bayesian Optimization (2017)
6. Nguyen, N.V., Choi, S.M., Kim, W.S., Lee, J.W., Kim, S., Neufeld, D., Byun, Y.H.: Multidisciplinary unmanned combat air vehicle system design using multi-fidelity model. Aerospace Science and Technology (2013)
7. Zheng, L., Hedrick, T.L., Mittal, R.: A multi-fidelity modelling approach for evaluation and optimization of wing stroke aerodynamics in flapping flight. Journal of Fluid Mechanics (2013)
8. Pilania, G., Gubernatis, J.E., Lookman, T.: Multi-fidelity machine learning models for accurate bandgap predictions of solids. Computational Materials Science (2017)
9. Prieß, M., Koziel, S., Slawig, T.: Surrogate-based optimization of climate model parameters using response correction. Journal of Computational Science (2011)
10. Huang, D., Allen, T.T., Notz, W.I., Zeng, N.: Global optimization of stochastic black-box systems via sequential kriging meta-models. Journal of global optimization (2006)
11. Xu, J., Zhang, S., Huang, E., Chen, C.H., Lee, L.H., Celik, N.: Mo2tos: Multi-fidelity optimization with ordinal transformation and optimal sampling. Asia-Pacific Journal of Operational Research (2016)

12. Yong, H.K., Wang, L., Toal, D.J., Keane, A.J., Stanley, F.: Multi-fidelity kriging-assisted structural optimization of whole engine models employing medial meshes. Structural and Multidisciplinary Optimization (2019)
13. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in NLP. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
14. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. In: Neural Information Processing Systems (2012)
15. Kennedy, M.C., O'Hagan, A.: Predicting the output from a complex computer code when fast approximations are available. Biometrika (2000)
16. Lam, R., Allaire, D.L., Willcox, K.E.: Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In: Structures, Structural Dynamics, and Materials Conference (2015)
17. Klein, A., Falkner, S., Bartels, S., Hennig, P., Hutter, F.: Fast Bayesian optimization of machine learning hyperparameters on large datasets. International Conference on Artificial Intelligence and Statistics (2017)
18. Kandasamy, K., Dasarathy, G., Oliva, J.B., Schneider, J., Póczos, B.: Gaussian process bandit optimisation with multi-fidelity evaluations. In: Neural Information Processing Systems (2016)
19. Kandasamy, K., Dasarathy, G., Schneider, J., Póczos, B.: Multi-fidelity Bayesian optimisation with continuous approximations. In: International Conference in Machine Learning (2017)
20. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence (1995)
21. McLeod, M., Osborne, M.A., Roberts, S.J.: Practical bayesian optimization for variable cost objectives. arXiv preprint arXiv:1703.04335 (2017)
22. Poloczek, M., Wang, J., Frazier, P.: Multi-information source optimization. In: Neural Information Processing Systems (2017)
23. Wu, J., Toscano-Palmerin, S., Frazier, P.I., Wilson, A.G.: Practical multi-fidelity bayesian optimization for hyperparameter tuning. arXiv preprint arXiv:1903.04703 (2019)
24. Picheny, V., Ginsbourger, D., Richet, Y., Caplin, G.: Quantile-based optimization of noisy computer experiments with tunable precision. Technometrics (2013)
25. Wang, Z., Jegelka, S.: Max-value entropy search for efficient Bayesian optimization. In: International Conference on Machine Learning (2017)
26. Takeno, S., Fukuoka, H., Tsukada, Y., Koyama, T., Shiga, M., Takeuchi, I., Karasuyama, M.: Multi-fidelity bayesian optimization with max-value entropy search. arXiv preprint arXiv:1901.08275 (2019)
27. Rasmussen, C.E.: Gaussian processes in machine learning. In: Advanced Lectures on Machine Learning, pp. 63–71. Springer (2004)
28. Álvarez, M.A., Lawrence, N.D.: Computationally efficient convolved multiple output Gaussian processes. Journal of Machine Learning Research (2011)
29. Cover, T.M., Thomas, J.A.: Elements of information theory. John Wiley & Sons (2012)
30. Azzalini, A.: A class of distributions which includes the normal ones. Scandinavian Journal of Statistics (1985)
31. Arnold, B.C., Beaver, R.J., Groeneveld, R.A., Meeker, W.Q.: The nontruncated marginal of a truncated bivariate normal distribution. Psychometrika (1993)
32. Arellano-Valle, R.B., Contrera-Reyes, J.E., Genton, M.G.: Shannon entropy and mutual information for multivariate skew-elliptical distributions. Scandinavian Journal of Statistics (2013)

33. The GPyOpt authors: GPyOpt: A Bayesian optimization framework in Python. http://github.com/SheffieldML/GPyOpt (2016)
34. Klein, A., Falkner, S., Mansur, N., Hutter, F.: RoBo: A flexible and robust Bayesian optimization framework in Python. In: Neural Information Processing Systems: Bayesian Optimization Workshop (2017)
35. Paleyes, A., Pullin, M., Mahsereci, M., Lawrence, N., Gonzlez, J.: Emulation of physical processes with emukit. In: Second Workshop on Machine Learning and the Physical Sciences, NeurIPS (2019)
36. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the lipschitz constant. Journal of optimization Theory and Applications (1993)
37. Moss, H., Leslie, D., Rayson, P.: Using J-K-fold cross validation to reduce variance when tuning NLP models. In: International Conference on Computational Linguistics (2018)
38. Moss, H., Moore, A., Leslie, D., Rayson, P.: FIESTA: Fast IdEntification of State-of-The-Art models using adaptive bandit algorithms. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
39. Xiong, S., Qian, P.Z., Wu, C.J.: Sequential design and analysis of high-accuracy and low-accuracy computer codes. Technometrics (2013)
40. Forrester, A., Sobester, A., Keane, A.: Engineering design via surrogate modelling: a practical guide. John Wiley & Sons (2008)
41. Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., Leyton-Brown, K.: Towards an empirical foundation for assessing Bayesian optimization of hyperparameters. In: Neural Information Processing Systems: Workshop on Bayesian Optimization (2013)
42. Deng, L.: The MNIST database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine (2012)
43. Siebert, J.: Vehicle recognition using rule based methods. Turing Institute,Glasgow (1987)
44. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Association for computational linguistics (2011)
45. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Neural Information Processing Systems (2008)
46. Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent Dirichlet allocation. In: Neural Information Processing Systems (2010)
47. Bengio, Y., Grandvalet, Y.: No unbiased estimator of the variance of k-fold cross-validation. Journal of Machine Learning Research (2004)

6 Supplementary Material for MUMBO

A Calculation of the MUMBO acquisition function

We now provide a thorough description of our proposed approach to calculate the MUMBO acquisition function for any choice of $\mathbf{x}$ and $\mathbf{z}$:

$$\alpha_n^{MUMBO}(\mathbf{x},\mathbf{z}) = H(y(\mathbf{x},\mathbf{z}) \,|\, D_n) - \mathbb{E}_{g^*}[H(y(\mathbf{x},\mathbf{z}) \,|\, g^*, D_n)]\,. \tag{6}$$

For ease of notation we drop the dependence on $\mathbf{x}$ and $\mathbf{z}$, so that g denotes the target function value at $\mathbf{x}$, f denotes the evaluation of $\mathbf{x}$ at fidelity $\mathbf{z}$, and y denotes the (noisy) observed value of $f(\mathbf{x},\mathbf{z})$, and seek to calculate the respective acquisition value α_n^{MUMBO}. From our underlying GP model we can extract our current beliefs about g and f as following a bi-variate Gaussian distribution:

$$\begin{pmatrix} g \\ f \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_g \\ \mu_f \end{pmatrix}, \begin{pmatrix} \sigma_g^2 & \Sigma \\ \Sigma & \sigma_f^2 \end{pmatrix}\right].$$

Then, noting that $Cov(y,g) = \Sigma$, we can write a similar expression for our current beliefs about g and noisy observations y as

$$\begin{pmatrix} g \\ y \end{pmatrix} \sim N\left[\begin{pmatrix} \mu_g \\ \mu_f \end{pmatrix}, \begin{pmatrix} \sigma_g^2 & \Sigma \\ \Sigma & \sigma_f^2 + \sigma^2 \end{pmatrix}\right].$$

We now derive analytical expressions for these predictive distributions from our underlying GP model. We denote our chosen kernel (defined over $\mathcal{X}\times\mathcal{Z}$) as k, so that $k((\mathbf{x},\mathbf{z}),(\mathbf{x}',\mathbf{z}'))$ represents our prior co-variance between the evaluation of $\mathbf{x}$ on fidelity $\mathbf{z}$ and the evaluation of $\mathbf{x}'$ on fidelity $\mathbf{z}'$. Denote the location in the fidelity space that corresponds to the true objective function as $\mathbf{z}_0$ (i.e. $f(\mathbf{x},\mathbf{z}_0) = g(\mathbf{x})$). For observations D_n, let $\mathbf{y}_n$ be the observed y values, define the kernel matrix $\mathbf{K}_n = [k((\mathbf{x}_i,\mathbf{z}_i),(\mathbf{x}_j,\mathbf{z}_j))]_{(\mathbf{x}_i,\mathbf{z}_i),(\mathbf{x}_j,\mathbf{z}_j)\in D_n}$ and kernel vectors $\mathbf{k}_n((\mathbf{x},\mathbf{z})) = [k((\mathbf{x}_i,\mathbf{z}_i),(\mathbf{x},\mathbf{z}))]_{(\mathbf{x}_i,\mathbf{z}_i)\in D_n}$. Then, following [27], the terms of our bi-variate Gaussian after observations D_n are:

$$\begin{aligned}
\mu_g &= \mathbf{k}_n((\mathbf{x},\mathbf{z}_0))^T(\mathbf{K}_n + \sigma^2 I)^{-1}\mathbf{y}_n \\
\mu_f &= \mathbf{k}_n((\mathbf{x},\mathbf{z}))^T(\mathbf{K}_n + \sigma^2 I)^{-1}\mathbf{y}_n \\
\sigma_g^2 &= k((\mathbf{x},\mathbf{z}_0),(\mathbf{x},\mathbf{z}_0)) - \mathbf{k}_n((\mathbf{x},\mathbf{z}_0))^T(\mathbf{K}_n + \sigma^2 I)^{-1}\mathbf{k}_n((\mathbf{x},\mathbf{z}_0)) \\
\sigma_f^2 &= k((\mathbf{x},\mathbf{z}),(\mathbf{x},\mathbf{z})) - \mathbf{k}_n((\mathbf{x},\mathbf{z}))^T(\mathbf{K}_n + \sigma^2 I)^{-1}\mathbf{k}_n((\mathbf{x},\mathbf{z})) \\
\Sigma &= k((\mathbf{x},\mathbf{z}),(\mathbf{x},\mathbf{z}_0)) - \mathbf{k}_n((\mathbf{x},\mathbf{z}))^T(\mathbf{K}_n + \sigma^2 I)^{-1}\mathbf{k}_n((\mathbf{x},\mathbf{z}_0))).
\end{aligned}$$

Following the advice of [14] we consistently use a Matérn 5/2 kernel to model performance surfaces over the hyper-parameter space.

The first term of (6) is simply the differential entropy of a Gaussian distribution and so can be calculated analytically as $\frac{1}{2}\log(2\pi e(\sigma_f^2 + \sigma^2))$. The second term of (4) is an expectation over the maximum value of the true objective g^*, which can be approximated using a Monte Carlo approach; we use [25]'s method

to approximately sample from $g^* \mid D_n$ using a mean-field approximation and extreme value theory, generating a set of N values $G = \{g_1, \ldots, g_N\}$. For each d-dimensional example in Section 4, we base our mean-field approximation on a grid of GP predictions at $10,000d$ random locations and any already evaluated locations. Note that we generate only one set of N samples of g^* for each BO step and all the required acquisition queries in that step are calculated with respect to this sample.

All that remains is to calculate the quantity inside the expectation for a given value of g^*, i.e the differential entropy of the random variable $y|g < g^*$ with a distribution that we now derive.

A.1 Derivation of the Extended Skew Normal Distribution

To simplify notation, rather than manipulating the co-variance Σ directly, we derive MUMBO in terms of the predictive correlation between y and g:

$$\rho = \frac{\Sigma}{\sigma_g \sqrt{\sigma_f^2 + \sigma^2}}.$$

Then using the well-known result for the conditional distribution of a bi-variate normal, we know that $g \mid y$ is also normally distributed with mean $\mu_g + \frac{\sigma_g}{\sqrt{\sigma_f^2+\sigma^2}}\rho(y - \mu_f)$ and variance $\sigma_g^2(1-\rho^2)$. We can therefore write the cumulative distribution function for $y|g \leq g^*$ as

$$\begin{aligned}\mathbb{P}(y \leq \theta | g \leq g*) =& \frac{\mathbb{P}(y \leq \theta, g \leq g*)}{\mathbb{P}(g \leq g*)} \\ =& \frac{\int_{-\infty}^{\theta} \phi(\frac{u-\mu_f}{\sqrt{\sigma_f^2+\sigma^2}}) \Phi\left(\frac{g* - \mu_g - \frac{\sigma_g}{\sqrt{\sigma_f^2+\sigma^2}}\rho(u-\mu_f)}{\sqrt{\sigma_g^2(1-\rho^2)}}\right) du}{\sqrt{\sigma_f + \sigma^2}\Phi(\frac{g*-\mu_g}{\sigma_g})}.\end{aligned}$$

After differentiating with respect to θ and defining $\gamma_{g^*} = \frac{g^*-\mu_g}{\sigma_g}$, we can write down the probability density function for the standardized variable $Z_{g^*} = \frac{y-\mu_f}{\sqrt{\sigma_f^2+\sigma^2}} | g < g^*$ as;

$$p(\theta) = \frac{1}{\Phi(\gamma_{g^*})}\phi(\theta)\Phi\left(\frac{\gamma_{g^*} - \rho\theta}{\sqrt{1-\rho^2}}\right),$$

which we recognize as the density of an extended skew normal distribution (ESG) [30], with moments

$$\mathbb{E}(Z_{g^*}) = \rho\frac{\phi(\gamma_{g^*})}{\Phi(\gamma_{g^*})}, \quad \mathrm{Var}(Z_{g^*}) = 1 - \rho^2\frac{\phi(\gamma_{g^*})}{\Phi(\gamma_{g^*})}\left[\gamma_{g^*} + \frac{\phi(\gamma_{g^*})}{\Phi(\gamma_{g^*})}\right]. \tag{7}$$

As [32] show that the differential entropy of an ESG is non-analytical, so too must be the final term in our MUMBO acquisition (6). We therefore perform numerical integration using Simpson's rule across eight standard deviations around the mean of Z_{g^*} (quantities provided by (7)). Note that the equivalent quantity in the original implementation of MES (without fidelity considerations) has a truncated normal distribution, with a closed form expression for its entropy.

A.2 Derivation of the full MUMBO acquisition function

We can now derive the simplified form (5) of the MUMBO acquisition function presented in Section 3, starting from the information-theoretic definition (6). Noting that $H(y|g^*, D_n) = H(Z_{g^*}) + \frac{1}{2}\log(\sigma_f^2 + \sigma^2)$ and that $H(y|D_n) = \frac{1}{2}\log(2\pi e(\sigma_f^2 + \sigma^2))$, we can rewrite (6) for a fixed choice of $\mathbf{x}$ and $\mathbf{z}$ as

$$\alpha_n^{MUMBO} = \frac{1}{2}\log(2\pi e) - \mathbb{E}_{g^*}\left[H(Z_{g^*})\right].$$

The differential entropy $H(Z_{g^*})$ for a fixed sample g^* can be decomposed into three terms

$$H(Z_{g^*}) = \mathbb{E}_{\theta\sim Z_{g^*}}\left[-\log(\phi(\theta)) + \log(\Phi(\gamma_{g^*})) - \log\left(\Phi\left(\frac{\gamma_{g^*} - \rho\theta}{\sqrt{1-\rho^2}}\right)\right)\right]$$

After expanding the first of these terms as

$$\mathbb{E}_{\theta\sim Z_{g^*}}\left[-\log(\phi(\theta))\right] = \tfrac{1}{2}\mathbb{E}_{\theta\sim Z_{g^*}}\left[\theta^2\right] + \tfrac{1}{2}\log(2\pi),$$

and further expanding using our expressions for the moments of Z_{g^*}, we now have

$$\alpha_n^{MUMBO} = \mathbb{E}_{g^*}\left[\rho^2\frac{\gamma_{g^*}\phi(\gamma_{g^*})}{2\Phi(\gamma_{g^*})} - \log(\Phi(\gamma_{g^*})) + \mathbb{E}_{\theta\sim Z_{g^*}}\left[\log\left(\Phi\Big\{\frac{\gamma_{g^*} - \rho\theta}{\sqrt{1-\rho^2}}\Big\}\right)\right]\right].$$

Therefore, after reintroducing dependence on $\mathbf{x}$ and $\mathbf{z}$ and replacing the expectation over g^* with a Monte-Carlo approximation across our set of N samples G, we see that MUMBO can be expressed as

$$\begin{aligned}\alpha_n^{MUMBO}(\mathbf{x},\mathbf{z}) \approx \frac{1}{N}\sum_{g^*\in G}\Bigg[&\rho(\mathbf{x},\mathbf{z})^2\frac{\gamma_{g^*}(\mathbf{x})\phi(\gamma_{g^*}(\mathbf{x}))}{2\Phi(\gamma_{g^*}(\mathbf{x}))} - \log(\Phi(\gamma_{g^*}(\mathbf{x})))\\ &+ \mathbb{E}_{\theta\sim Z_{g^*}(\mathbf{x},\mathbf{z})}\left[\log\left(\Phi\Big\{\frac{\gamma_{g^*}(\mathbf{x}) - \rho(\mathbf{x},\mathbf{z})\theta}{\sqrt{1-\rho^2(\mathbf{x},\mathbf{z})}}\Big\}\right)\right]\Bigg].\end{aligned}$$

B Experiment Details

We now provide implementation details for our all our experiments.

B.1 Discrete Multi-fidelity BO

Figure 2 shows the performance of MUMBO over the standard MF benchmark functions used by [39] and [18]. These problems have an objective function and a discrete hierarchy of low-fidelity approximations that can be queried at reduced cost. We measure the performance of the MF approaches in terms of the total resources spent on query costs after random initializations. We wish to find high-performing incumbents after spending few resources. We generate starting points for the optimization by querying twice as many random points as the problem dimension and evaluate these across all fidelities. For the information-theoretic approaches we also provide the time spent deciding where to make each successive evaluation as this is an important practical consideration.

In Figure 2 we present the performance of the MF-GP-UCB algorithm of [19] using their published code. Unfortunately we were unable to achieve performance on these functions even close to the level claimed in their work. However, our approaches outperform even the results claimed in their paper. This performance discrepancy is likely due to our different initialization scheme and that we do not tune their algorithm's hyper-parameters (illustrating the benefit of using a parameter-free approach like MUMBO). Also note that MF-GP-UCB models fidelities as separate GPs, whereas MUMBO and MTBO use the more sophisticated coregionaliazation model.

We now provide detailed information about each of our synthetic functions.

Forrester Function. A single dimensional function [40] defined on $\mathcal{X} = [0, 1]$ with three fidelitlies with query costs 10, 5 and 2:

$$\begin{aligned} f(x_1, 0) &= (6x_1 - 2)^2 \sin(12x_1 - 4) \\ f(x_1, 1) &= 0.75 f(x_1, 0) + 3(x_1 - 0.5) + 2 \\ f(x_1, 2) &= 0.5 f(x_1, 0) + 5(x_1 - 0.5) + 2 \end{aligned}$$

Currin exponential function (discete fidelity space). A two-dimensional function defined on $\mathcal{X} = [0, 1]^2$ with two fidelities queried with costs 10 and 1:

$$\begin{aligned} f(x_1, x_2, 0) =& \left(1 - \exp(-\frac{1}{2x_2})\right) \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20} \\ f(x_1, x_2, 1) =& \frac{1}{4} f(x_1 + 0.05, x_2 + 0.05, 0) \\ &+ \frac{1}{4} f(x_1 + 0.05, max(0, x_2 - 0.05), 0) \\ &+ \frac{1}{4} f(x_1 - 0.05, x_2 + 0.05, 0) \\ &+ \frac{1}{4} f(x_1 - 0.05, max(0, x_2 - 0.05), 0). \end{aligned}$$

Hartmann 3 function. A three-dimensional function with 4 local extrema defined on $\mathcal{X} = [0, 1]^3$ with three fidelities ($m = 0, 1, 2$) queried at costs $100, 10$ and 1:

$$f(x_1, x_2, x_3, m) = -\sum_{i=1}^{4} \alpha_{i,m+1} \exp\left(-\sum_{j=1}^{3} A_{i,j}(x_j - P_{i,j})^2\right),$$

where

$$A = \begin{pmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}, \quad \alpha = \begin{pmatrix} 1 & 1.01 & 1.02 \\ 1.2 & 1.19 & 1.18 \\ 3 & 2.9 & 2.8 \\ 3.2 & 3.3 & 3.4 \end{pmatrix}, \quad P = \begin{pmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{pmatrix}.$$

Hartmann 6 function. A six-dimensional function defined on $\mathcal{X} = [0, 1]^6$ with four fidelities ($m = 0, 1, 2, 3$) queried at costs $1000, 100, 10$ and 1:

$$f(x_1, x_2, x_3, x_4, x_5, x_6, m) = -\sum_{i=1}^{4} \alpha_{i,m+1} \exp\left(-\sum_{j=1}^{6} A_{i,j}(x_j - P_{i,j})^2\right),$$

where

$$A = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}, \qquad \alpha = \begin{pmatrix} 1 & 1.01 & 1.02 & 1.03 \\ 1.2 & 1.19 & 1.18 & 1.17 \\ 3 & 2.9 & 2.8 & 2.7 \\ 3.2 & 3.3 & 3.4 & 3.5 \end{pmatrix},$$

$$P = \begin{pmatrix} 1312 & 1696 & 5569 & 124 & 8283 & 5886 \\ 2329 & 4135 & 8307 & 3736 & 1004 & 9991 \\ 2348 & 1451 & 3522 & 2883 & 3047 & 6650 \\ 4047 & 8828 & 8732 & 5743 & 1091 & 381 \end{pmatrix}.$$

Borehole function. An eight-dimensional function defined on

$$\begin{aligned} \mathcal{X} = [&0.05, 0.15; 100, 50,000; 63070, 115600; 990, \\ &1110; 63.1, 116; 700, 820; 1120, 1680; 9855, 12055] \end{aligned}$$

with two fidelities queried with costs 10 and 1:

$$\begin{aligned} f(\mathbf{x}, 0) =& \frac{2\pi x_3(x_4 - x_6)}{\log(x_2/x_1)\left(1 + \frac{2x_7 x_3}{log(x_2/x_1)x_1^2 x_8} + \frac{x_3}{x_5}\right)}, \\ f(\mathbf{x}, 1) =& \frac{5x_3(x_4 - x_6)}{\log(x_2/x_1)\left(1.5 + \frac{2x_7 x_3}{log(x_2/x_1)x_1^2 x_8} + \frac{x_3}{x_5}\right)}. \end{aligned}$$

B.2 Continuous Multi-fidelity BO: FABOLAS

For our second set of experiments, we consider the MF hyper-parameter tuning framework of [17], which dynamically chooses the amount of training data used for hyper-parameter evaluations. Their FABOLAS algorithm is widely regarded as state-of-the-art, achieving hyper-parameter tuning with orders of magnitude less computation that standard BO and other competing MF tuning routines. We use the code provided for FABOLAS within the ROBO package [34] by the same authors. We use their implementation exactly, only swapping out their original MTBO acquisition function for our proposed MUMBO acquisition. A good hyper-parameter tuner finds hyper-parameter configurations that will perform well on new data after using as little computational resource as possible, including effort spent fitting models and deciding the hyper-parameter configuration and fidelity to query. By splitting our data into train, validation and test sets, we are able to report total wall-clock time against the performance (in accuracy) of incumbents on this test set (calculated retrospectively at the end of the optimization). During the optimization, models are trained on random subsets of chosen proportions of the training set and tested on the full validation set.

We consider the same examples as [17], using the same data-sets downloaded from the HPOlib BO benchmark repository [41] of MNIST [42] and Vehicle Registrations [43] - we refer the reader to their work for specific details. As a result of limited computational resources and wishing to repeat each experiment over multiple random seeds, we had to halve the training data (to $25,000$ for both MNIST and Vehicle) throughout the experiment (including testing the incumbents). We do, however, use the full test and validation sets. For each replication, we start with 10 random hyper-parameter initializations each evaluated on $\frac{1}{64}$, $\frac{1}{32}$, $\frac{1}{16}$ and $\frac{1}{8}$ of the training data.

B.3 Multi-task BO: FASTCV

In Section 4.4, we test MUMBO in a multi-task framework by providing the first information-theoretic implementation of FASTCV [4], where we sequentially make evaluations on a single K-fold CV folds with the aim of optimizing the evaluations based on all K folds. As discussed in Section 4.4, the original implementation of FASTCV chooses hyper-parameters to evaluate and then the fold upon which to make the evaluation as a two-stage heuristic based on the expected improvement acquisition. In Figure 4, we investigate the change in performance of replacing this acquisition function with the principled MT decision-making provided by our MUMBO acquisition function. We also present the performance of standard BO routines that have to evaluate all K CV folds for each hyper-parameter query. For ML models, the acquisition function overheads are insignificant compared to the costs of fitting the model on large proportions of the training data (unlike the small proportions chosen by FABOLAS), and so we measure the performance of our algorithms by the number of individual model fits required to reach a certain incumbent performance. To allow the fair

comparison of the computational resources used by each algorithm, we consider a single optimization step as the evaluation of a single model on a single fold and so each hyper-parameter evaluation using K-fold CV counts as K steps.

We consider two well-known ML tasks: using a support vector machine (SVM) to classify the sentiment in IMDB movie reviews [44] and using probabilistic matrix factorization (PMF) [45] to recommend movies on the Movielens-100k data set [46]. We tune the regularization strength across $[e^{-5}, e^{25}]$ and kernel coefficient across $[e^{-25}, e^{5}]$ for the SVM and the learning rate across $[0, 0.01]$, regularization strength across $[0, 0.1]$, matrix rank across $[50, .., 150]$ and number of model epochs across $[10, .., 50]$ for the PMF. To create a difficult MT optimization problem, we use only a small subset of the IMDB data (a random subset of $1,000$ reviews split into 10 folds) as this increases the between-fold variability of a K-fold CV estimate [47] and so limits the similarity of evaluations on different folds that is exploited by FASTCV. Despite this challenging MT set-up, both the original FASTCV and MUMBO are able to provide significantly faster tuning than standard approaches, with MUMBO providing an additional increase in test performance over FASTCV (as based on the reliable performance estimates calculated on the $49,000$ reviews not used for training). In addition, we also consider the whole of the large Movelens-100k dataset split into 5 folds. Despite the stochastic nature of PMF meaning that our tuning algorithms have deal with high levels of observation noise for each hyper-parameter evaluation, we once again we see that the principled decision-making of MUMBO allows much faster optimization than all the other approaches - achieving lower 5-fold CV estimated mean squared error (a standard measurement of performance for recommendation systems).

B.4 Wider Comparison With Existing Methods

We now present the functions used for final experiments.

Currin exponential function (continuous fidelity space). A two-dimensional function defined on $\mathcal{X} = [0, 1]^2$ with fidelity space $z \in [0, 1]$. The cost of querying fidelity z is given by $\lambda(z) = 0.1 + z^2$ with the objective lying at fidelity $z = 1$.

$$f(x_1, x_2, z) = \left(1 - 0.1(1 - z)\exp(-\frac{1}{2x_2})\right)\frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20}.$$

Rosenbrock function. A two-dimensional function defined on $\mathcal{X} = [-2, 2]^2$ with two fidelities ($m = 0, 1$) queried at costs 1000 and 1. Observations are contaminated with Gaussian noise with variance 0.001 and $1e{-}6$ for each fidelity respectively.

$$\begin{aligned} f(x_1, x_2, 0) =& (1 - x_1)^2 + 100(x_2 - x_1^2)^2 \\ f(x_1, x_2, 1) =& f(x_1, x_2, 0) + 0.1\sin(10x_1 + 5x_2) \end{aligned}$$