Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 407 (2023) 115937
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Multi-fidelity cost-aware Bayesian optimization

Zahra Zanjani Foumani, Mehdi Shishehbor, Amin Yousefpour, Ramin Bostanabad*

Department of Mechanical and Aerospace Engineering, University of California Irvine, Irvine, CA, United States of America

Received 4 November 2022; received in revised form 1 February 2023; accepted 1 February 2023
Available online xxxx

Dataset link: GitLab repository, https://gitlab.com/S3anaz/multi-fidelity-cost-aware-bayesian-optimization/-/tree/main/Notebooks

Abstract

Bayesian optimization (BO) is increasingly employed in critical applications such as materials design and drug discovery. An increasingly popular strategy in BO is to forgo the sole reliance on high-fidelity data and instead use an ensemble of information sources which provide inexpensive low-fidelity data. The overall premise of this strategy is to reduce the total sampling costs by querying inexpensive low-fidelity sources whose data are correlated with high-fidelity samples. Here, we propose a multi-fidelity cost-aware BO framework that dramatically outperforms the state-of-the-art technologies in terms of efficiency, consistency, and robustness. We demonstrate the advantages of our framework on analytic and engineering problems and argue that these benefits stem from our two main contributions: (1) we develop a novel acquisition function for multi-fidelity cost-aware BO that safeguards the convergence against the biases of low-fidelity data, and (2) we tailor a newly developed emulator for multi-fidelity BO which enables us to not only simultaneously learn from an ensemble of multi-fidelity datasets, but also identify the severely biased low-fidelity sources that should be excluded from BO.

Keywords: Bayesian optimization; Multi-fidelity modeling; Emulation; Resource allocation; Manifold learning; Gaussian process

1. Introduction and related works

Bayesian optimization (BO) is an iterative and sample-efficient global optimization technique that has been successfully applied to a wide range of applications including materials discovery [1–4], design of chemical systems such as catalysts [5], hyperparameter tuning in machine learning (ML) models [6], robot motion control [7], and updating internet-scale software systems [8].

While BO is very effective, the total optimization cost can still be high if only an expensive source is sampled during the optimization (e.g., experiments or costly simulations). To reduce the overall data collection costs in such scenarios an increasingly popular strategy is to formulate *multi-fidelity* (MF) methods that use multiple data sources which typically have different levels of accuracy and cost, see [9–12] for some applications. Assuming low-fidelity (LF) sources are cheaper to query, the overall premise of these methods is to reduce the total sampling costs by leveraging the correlations between low- and high-fidelity (HF) data. In this paper, we propose a *multi-fidelity cost-aware (MFCA) BO framework* that optimizes an expensive objective function using an ensemble of data sources with

* Corresponding author.
E-mail address: Raminb@uci.edu (R. Bostanabad).

https://doi.org/10.1016/j.cma.2023.115937
0045-7825/

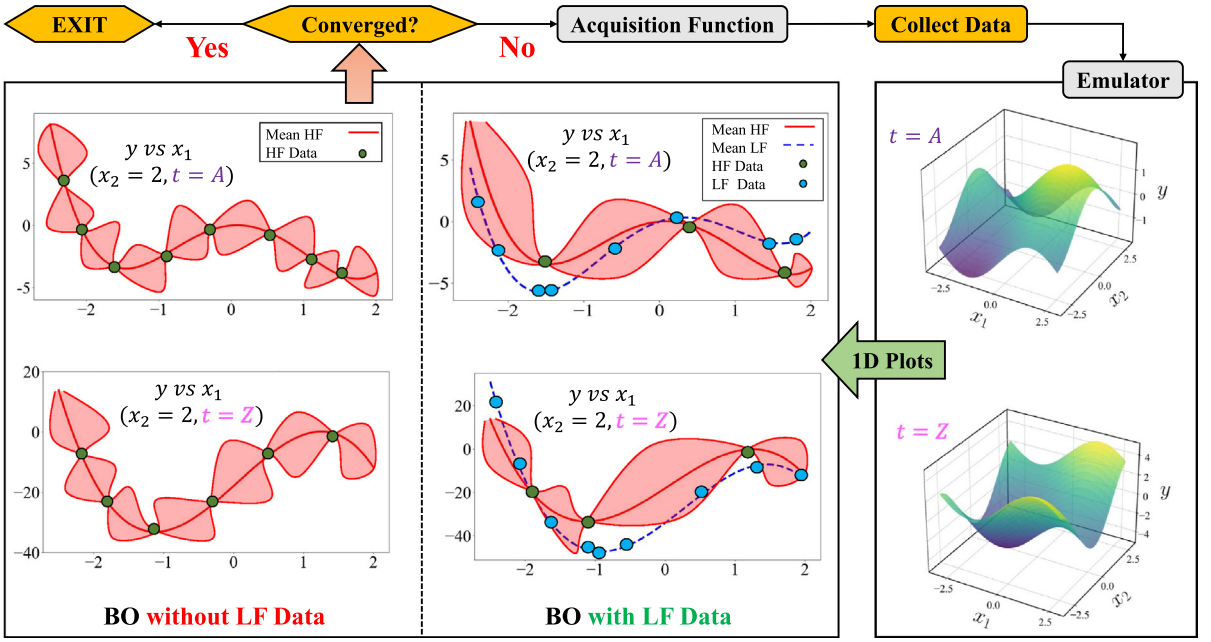


Fig. 1. Bi-fidelity BO: We illustrate the overall flow for optimizing a function with two numerical inputs (x_1, x_2) and one categorical input (t) that has two levels (i.e., $t = A$ or $t = Z$). The figure also considers both single-fidelity (without LF data) and bi-fidelity (with LF data) optimization scenarios. In the left box, the dashed line separates the two optimization scenarios. The upper and lower plots on each side of the dashed line visualize the function in $1D$. For example, in the bi-fidelity case (with LF data), we visualize the emulation for HF and LF as a function of x_1 for either $t = A$ (top plot) or $t = Z$ (bottom plot). Note that (1) we have fixed x_2 to 2 so that we can draw in $1D$, and (2) t is categorical. In both scenarios, the first step is training the emulator on the data (right box). Then, the emulator is passed to the left box for optimization. In the optimization process of the SF model (without LF data), the optimum is found by merely sampling from the expensive HF source. However, in the MF scenario (with LF data), the LF source is also used in sampling which is correlated with the HF source and hence reduces the reliance on HF data. This iterative process of sampling-emulation is repeated until the convergence condition is met.

arbitrary levels of accuracy and cost. We provide a new perspective on probabilistic learning from multiple sources which endows our framework with four major advantages over existing MF BO techniques: (a) safeguarding the convergence against the biases of the LF sources even if they are extremely inexpensive to query (i.e., if the majority of the samples are LF), (b) learning the relative fidelity of the sources rather than requiring a priori determination of such relations by the user, (c) dispensing with the assumptions that aim to relate the fidelity and cost of a data source, and (d) improving numerical stability and efficiency.

As schematically demonstrated in Fig. 1 and detailed in Section 2.2, BO has two main ingredients that interact sequentially to optimize a black-box and expensive-to-evaluate objective function. These two ingredients include an emulator (i.e., a probabilistic surrogate) and an acquisition function (AF). The optimization process starts by fitting the emulator to a small initial dataset that is typically obtained via design of experiments. The emulator is next used in the AF to determine the candidate input(s) whose corresponding output(s) must be obtained by querying the expensive function. Given the new sample(s), the training dataset is updated and the entire fitting-searching-sampling process is repeated until a convergence criterion is met (e.g., resources are exhausted).

While many emulators such as Bayesian neural networks are available, Gaussian processes (GPs) are typically used in BO since they very efficiently learn from small data, are easy and fast to train, provide prediction uncertainties, and have interpretable parameters [13–16]. Use of GPs in BO has increased even more because of the recent works that enable them to handle categorical variables [17–19], high-dimensional inputs [20,21], large datasets [22–25], and non-stationary noise [26,27]. As for the AF, there are many choices [28,29] such as expected improvement (EI), probability of improvement (PI), and knowledge gradient (KG). The primary difference among these AFs is that they select the candidate input(s) while taking different approaches for balancing exploitation (i.e., sampling based on the best predictions of the emulator) and exploration (i.e., sampling to reduce prediction

uncertainty). This selection involves integration which can sometimes be analytically computed (e.g., when GP and EI are chosen as the emulator and AF, respectively).

To optimally use an ensemble of information sources in BO, two conditions must be met: (1) the emulator should leverage the cross-source correlations (which are hidden in the datasets) to more accurately surrogate *all* the data sources (esp. the HF one), and (2) the AF should appropriately calculate the value or *utility* of a to-be-sampled data point based on its source and evaluation cost. Satisfying these two conditions in many realistic applications is nontrivial for the following reasons:

- The global optima (input and output) of LF sources differ from those of the HF source, see Fig. 4 for a one dimensional illustration.
- Some LF sources have major biases (which are a priori unknown to the analyst) and must be excluded from the search process from the very beginning. Including such LF sources increases the overall sampling costs and may also result in convergence to an incorrect solution.
- If highly cheap LF sources are available, a naively designed AF chooses to sample from them very frequently since the information value of a candidate point is inversely scaled by the cost of its source. This heavily biased sampling can force BO to converge to the optima of those sources rather than the HF source.

As reviewed in Section 2.2, existing multi-fidelity BO technologies partially address these challenges by ad hoc tuning and making simplifying assumptions. These assumptions often include presuming a direct relationship between fidelity level and sampling cost, assuming LF sources are always useful, or manually adjusting the sampling costs (e.g., converting the $1000/50/1$ cost ratio between three sources to $1/0.5/0.1$). These manual changes are quite laborious and result in either convergence to an incorrect solution or higher overall costs compared to single-fidelity (SF) BO which solely leverages the HF source.

In this paper, we provide new perspectives for learning from multi-fidelity sources in the context of BO. In particular, we argue that (a) the emulator must fuse the multi-source data in a nonlinearly learnt manifold to maximally leverage cross-source correlations and also indicate trustworthy LF sources that do not deteriorate BO's performance, and (b) the AF should use the available information on the LF sources solely for exploration and those on the HF sources for exploitation. As demonstrated in Section 4, these contributions endow our framework with significant performance improvements over existing technologies.

The rest of the paper is organized as follows. We review the relevant technical background in Section 2 and introduce our approach in Section 3 (readers familiar with GPs and MF BO can safely skip Section 2). We test the performance of our MFCA BO framework against the state-of-the-art on a set of analytic and two engineering problems in Section 4 and finally conclude the paper in Section 5. We also provide a nomenclature and list of symbols in Appendix H. The Gitlab repository associated with this project hosts supplementary materials.

2. Technical background

In this Section, we first provide some background on latent map Gaussian process (LMGP) which is one of the key components of our MFCA BO framework. Then, we elaborate on the two main ingredients of BO in Section 2.2 where we also review some of the existing methods for handling MF data.

2.1. Latent map Gaussian processes (LMGPs)

For metamodeling via GPs, one assumes that the training data comes from a multivariate normal distribution with parametric mean and covariance functions and then uses closed-form conditional distribution formulas for prediction. Below, we first detail the process for estimating these parameters when the input space contains categorical and/or numerical variables. Then, we provide the prediction formulas.

Assume the training data is a realization from a GP and that the following relation holds:

$$y(\boldsymbol{x}) = \boldsymbol{m}(\boldsymbol{x})\boldsymbol{\beta} + \xi(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_{dx}]^T$ is the input vector, $y(\boldsymbol{x})$ is the output/response, $\boldsymbol{m}(\boldsymbol{x}) = \left[m_1(\boldsymbol{x}), \ldots, m_{d\beta}(\boldsymbol{x})\right]$ are a set of parametric basis functions, $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_{d\beta}]^T$ are unknown coefficients, and $\xi(\boldsymbol{x})$ is a zero-mean GP whose covariance function or kernel is:

$$\operatorname{cov}\left(\xi(\boldsymbol{x}), \xi\left(\boldsymbol{x}'\right)\right) = c\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \sigma^2 r\left(\boldsymbol{x}, \boldsymbol{x}'\right) \tag{2}$$

where σ^2 is the variance of the process and $r(.,.)$ is a parametric correlation function. A common choice for $r(.,.)$ is the Gaussian correlation function:

$$r(\boldsymbol{x}, \boldsymbol{x}') = \exp\{-\sum_{i=1}^{dx} 10^{\omega_i}(x_i - x_i')^2\} = \exp\{-(\boldsymbol{x} - \boldsymbol{x}')^T 10^{\boldsymbol{\Omega}}(\boldsymbol{x} - \boldsymbol{x}')\} \tag{3}$$

where $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_{dx}]^T$ are the scale parameters and $\boldsymbol{\Omega} = \text{diag}(\boldsymbol{\omega})$. GP modeling largely depends on the choice of the correlation function which measures the weighted distance between any two inputs, see Eq. (3). As recently motivated in [30], to directly use GPs for MF modeling we must extend them such that they can handle categorical inputs. This extension primarily relies on reformulating $r(.,.)$ and can be accomplished in different ways. In this paper, we employ LMGPs [30] since (1) they have been shown to outperform other approaches, and (2) they provide a visualizable and interpretable manifold which can be used to detect discrepancies among data sources (this manifold helps us to exclude highly biased LF sources from BO).

Let us denote categorical inputs via $\boldsymbol{t} = [t_1, \ldots, t_{dt}]^T$ where variable t_i has l_i distinct levels. For example, $t_1 = \{Male, Female\}$ and $t_2 = \{Persian, American, Spanish\}$ are two categorical inputs where $l_1 = 2$ and $l_2 = 3$. To handle mixed inputs, $\boldsymbol{u} = [x_1, \ldots, x_{dx}, t_1, \ldots, t_{dt}]^T$, LMGP learns a unified parametric function[1] that maps each combination of categorical variables to a point in a quantitative manifold (aka latent space[2]). This mapping function can be incorporated into any standard correlation function (e.g., Gaussian, Matern, etc.) and the performance of LMGP is quite insensitive to this choice [31,32]. In this paper, we use the Gaussian correlation which is reformulated as follows for mixed inputs:

$$r(\boldsymbol{u}, \boldsymbol{u}') = \exp\{-(\boldsymbol{x} - \boldsymbol{x}')^T \boldsymbol{\Omega}(\boldsymbol{x} - \boldsymbol{x}') - \|\boldsymbol{z}(\boldsymbol{t}) - \boldsymbol{z}(\boldsymbol{t}')\|_2^2\} \tag{4}$$

or equivalently,

$$r(\boldsymbol{u}, \boldsymbol{u}') = \exp\{-\sum_{i=1}^{dx} 10^{\omega_i}(x_i - x_i')^2\} \times \exp\{-\sum_{i=1}^{dz}(z_i(\boldsymbol{t}) - z_i(\boldsymbol{t}'))^2\} \tag{5}$$

where $\|\cdot\|_2$ denotes the Euclidean 2-norm and $\boldsymbol{z}(\boldsymbol{t}) = [z_1(\boldsymbol{t}), \ldots, z_{dz}(\boldsymbol{t})]_{1\times dz}$ is the location in the learned latent space corresponding to the specific combination of the categorical variables denoted by $\boldsymbol{t}$. To find these latent points, LMGP first assigns a unique prior representation (a unique vector) to each combination of categorical variables. Then, it learns a linear transformation[3] that maps these unique vectors into a compact manifold with dimensionality dz:

$$\boldsymbol{z}(\boldsymbol{t}) = \boldsymbol{\zeta}(\boldsymbol{t})\boldsymbol{A} \tag{6}$$

where $\boldsymbol{t}$ denotes a specific combination of the categorical variables, $\boldsymbol{z}(\boldsymbol{t})$ is the $1 \times dz$ posterior latent representation of $\boldsymbol{t}$, $\boldsymbol{\zeta}(\boldsymbol{t})$ is the unique prior vector representation of $\boldsymbol{t}$, and $\boldsymbol{A}$ is a rectangular matrix that maps $\boldsymbol{\zeta}(\boldsymbol{t})$ to $\boldsymbol{z}(\boldsymbol{t})$. There are various methods for constructing the prior vectors $\boldsymbol{\zeta}$ and we refer the reader to [30] for more details. In this paper, we use grouped one-hot encoding which makes $\boldsymbol{\zeta}(\boldsymbol{t})$ and $\boldsymbol{A}$ to be of sizes $1 \times \sum_{i=1}^{dt} l_i$ and $\sum_{i=1}^{dt} l_i \times dz$, respectively. For instance, in the above example the grouped one-hot encoded version of the combination $\boldsymbol{t} = [Female, American]^T$ is $\boldsymbol{\zeta}(\boldsymbol{t}) = [0, 1, 0, 1, 0]$ where the first two numbers encode the levels of t_1 while the last three numbers indicate the levels of t_2.

To emulate via LMGP, the hyperparameters ($\boldsymbol{\beta}$, $\boldsymbol{A}$, $\boldsymbol{\omega}$, and σ^2) must be estimated via the training data. To find these estimates, we utilize maximum a posteriori (MAP) which estimates the hyperparameters such that they maximize the posterior of the n training data being generated by $y(\boldsymbol{x})$, that is:

$$[\widehat{\boldsymbol{\beta}}, \hat{\sigma}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{A}}] = \underset{\boldsymbol{\beta}, \sigma^2, \omega, \boldsymbol{A}}{\text{argmax}} \left|2\pi\sigma^2\boldsymbol{R}\right|^{-\frac{1}{2}} \times \exp\left\{\frac{-1}{2}(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\beta})^T \left(\sigma^2\boldsymbol{R}\right)^{-1}(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\beta})\right\} \times \underset{\boldsymbol{\beta}, \sigma^2, \omega, \boldsymbol{A}}{\text{P}(\cdot)} \tag{7}$$

or equivalently:

$$[\widehat{\boldsymbol{\beta}}, \hat{\sigma}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{A}}] = \underset{\boldsymbol{\beta}, \sigma^2, \omega, \boldsymbol{A}}{\text{argmin}} \frac{n}{2}\log\left(\sigma^2\right) + \frac{1}{2}\log(|\boldsymbol{R}|) + \frac{1}{2\sigma^2}(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\beta})^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\beta}) + \log(\underset{\boldsymbol{\beta}, \sigma^2, \omega, \boldsymbol{A}}{\text{P}(\cdot)}), \tag{8}$$

[1] Multiple mapping functions can also be used and we leverage this in Section 3.1.

[2] In a manifold or a latent space, high dimensional objects such as images are represented via low-dimensional quantitative features.

[3] More complex transformations based on, e.g., deep neural networks can also be used.

where $\log(\cdot)$ is the natural logarithm, $|\cdot|$ denotes the determinant operator, $\boldsymbol{y} = [y^1, \ldots, y^n]^T$ is the $n \times 1$ vector of outputs in the training data, $\boldsymbol{R}$ is the $n \times n$ correlation matrix with the (i, j)th element $R_{ij} = r(\boldsymbol{x}^i, \boldsymbol{x}^j)$ for $i, j = 1, \ldots, n$, $\boldsymbol{M}$ is the $n \times d\beta$ matrix with the (i, j)th element $M_{ij} = m_j(\boldsymbol{x}^i)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, d\beta$, and $P(\cdot)$ is the prior on the hyperparameters. In this paper, we place independent priors on the hyperparameters where $\sigma \sim Lognormal(0, 3)$ while $\boldsymbol{\omega}, \boldsymbol{\beta}, \boldsymbol{A}$ follow normal priors.[4] These priors are adopted in GPyTorch and shown to be effective [33].

The optimization problem in Eq. (8) can be efficiently solved via gradient-based optimization [16,34]. Once the hyperparameters are estimated, the conditional distribution formulas are used to predict the response distribution at the arbitrary point $\boldsymbol{x}^*$. The mean and variance of this normal distribution are:

$$\mathbb{E}[y(\boldsymbol{x}^*)] = \mu(\boldsymbol{x}^*) = \boldsymbol{m}(\boldsymbol{x}^*)\widehat{\boldsymbol{\beta}} + \boldsymbol{r}^T(\boldsymbol{x}^*)\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{M}\widehat{\boldsymbol{\beta}}) \tag{9}$$

$$\text{cov}(y(\boldsymbol{x}^*), y(\boldsymbol{x}^*)) = \sigma^2(\boldsymbol{x}^*) = \hat{\sigma}^2(1 - \boldsymbol{r}^T(\boldsymbol{x}^*)\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}^*) + \boldsymbol{g}(\boldsymbol{x}^*)(\boldsymbol{M}^T\boldsymbol{R}^{-1}\boldsymbol{M})^{-1}\boldsymbol{g}(\boldsymbol{x}^*)^T) \tag{10}$$

where $\mathbb{E}$ denotes expectation, $m(\boldsymbol{x}^*) = [m_1(\boldsymbol{x}^*), \ldots, m_{d\beta}(\boldsymbol{x}^*)]$, $\boldsymbol{r}(\boldsymbol{x}^*)$ is an $n \times 1$ vector with the ith element $r(\boldsymbol{x}^i, \boldsymbol{x}^*)$, and $\boldsymbol{g}(\boldsymbol{x}^*) = \boldsymbol{m}(\boldsymbol{x}^*) - \boldsymbol{M}^T\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}^*)$.

2.2. Bayesian optimization (BO)

BO is increasingly used to optimize expensive-to-evaluate (and typically black-box) functions. As opposed to gradient-based or heuristic optimization techniques that only rely on function evaluations (or predictions of a surrogate), BO leverages the probabilistic predictions of a sequentially updated emulator. In this paper, we focus on single-response functions over unconstrained and bounded domains but note that BO can also handle multi-response (or multi-task) [35], composite [36], or constrained problems [37].

As summarized in Algorithm 1, BO has an iterative nature where an emulator is first fitted to some initial training data. This emulator is then queried via the AF which calculates the expected utility of any point in the input space, i.e., $\mathbb{E}[I(\boldsymbol{x})]$ where $I(\boldsymbol{x})$ is the user-defined utility function. These queries are used in the auxiliary optimization problem[5] which aims to find the candidate point with the maximum expected utility in the input space. Once this point is found, the expensive function is queried and the resulting (input, output) pair is used to update the training data. Given the augmented dataset, the above process is repeated until a convergence metric is met.

Except for some special cases, solving the auxiliary optimization problem in Algorithm 1 is highly costly since each evaluation of its objective function (i.e., $\mathbb{E}[I(\boldsymbol{x})]$) amounts to integration which cannot be calculated analytically. Fortunately, the special cases perform quite well in most practical applications and hence are used frequently (we also employ them in our framework).

Algorithm 1 is strictly sequential in that the dataset is augmented with a single sample at each iteration. To leverage parallel computing, one can augment the dataset with a pool of samples which jointly maximize the expected utility [38]. Additionally, Algorithm 1 is myopic in that the AF does not consider the effect of the to-be-evaluated sample on the emulator in the future iterations. This myopic nature is addressed in look-ahead AFs such as KG (detailed below) which typically provide improved performance at the expense of significantly increasing the cost of solving the auxiliary optimization problem.

2.2.1. Single-fidelity acquisition functions

Many different AFs have been proposed for diverse applications and in this Section we review three of the most widely used ones: EI, PI, and KG. While the first two AFs are myopic,[6] KG is look-ahead. Any AF calculates the expected value of a user-defined utility function conditioned on the available data D, that is:

$$\alpha(\boldsymbol{x}) = \mathbb{E}[I(\boldsymbol{x}) \mid \mathcal{D}] \tag{11}$$

Our proposed AF for MFCA BO (see Section 3.2) leverages EI as well as PI and hence is also myopic. In Section 4 we compare our AF to EI, PI, and KG and indicate that it consistently outperforms them.

[4] $\boldsymbol{\omega} \sim \mathcal{N}(-3, 3)$, $\boldsymbol{\beta} \sim \mathcal{N}(0, 1)$, $\boldsymbol{A} \sim \mathcal{N}(0, 3)$.

[5] Gradient-based optimization techniques are almost always used at this stage.

[6] EI does have a look-ahead version [39].

Algorithm 1: Strictly Sequential and Myopic Bayesian Optimization for Maximization

Given: Initial data $\mathcal{D}^k = \{(\boldsymbol{x}^i, y^i)\}_{i=1}^k$, expensive black-box function $f(\boldsymbol{x})$
Define: Utility function $I(\boldsymbol{x})$, stop conditions
while *stop conditions not met* **do**
1. Train the GP emulator using $\mathcal{D}^k$
2. Define the acquisition function $\alpha(\boldsymbol{x}) = \mathbb{E}[I(\boldsymbol{x}) \mid \mathcal{D}^k]$
3. Solve the auxiliary optimization problem: $\boldsymbol{x}^{k+1} = \underset{\boldsymbol{x}\in\mathbb{X}}{\text{argmax}}\ \alpha(\boldsymbol{x})$
4. Query $f(\boldsymbol{x})$ at $\boldsymbol{x}^{k+1}$ to obtain y^{k+1}
5. Update data: $\mathcal{D}^{k+1} \leftarrow \mathcal{D}^k \cup (\boldsymbol{x}^{k+1}, y^{k+1})$
6. Update counter: $k \leftarrow k+1$

end
Output: Updated data $\mathcal{D}^k = \left\{\left(\boldsymbol{x}^i, y^i\right)\right\}_{i=1}^k$, GP emulator

PI is an AF that favors exploitation [40], i.e., it rewards samples that improve y^* which is the best function value seen so far. For instance, when maximizing the expensive black-box function $f(\boldsymbol{x})$, this AF uses the following utility function:

$$I_{PI}(\boldsymbol{x}) = \begin{cases} 1 & y(\boldsymbol{x}) > y^* \\ 0 & y(\boldsymbol{x}) \leq y^* \end{cases} \tag{12}$$

where $y(\boldsymbol{x})$ is the emulator-based prediction at $\boldsymbol{x}$. Based on Eq. (12), if $y(\boldsymbol{x})$ is less than y^*, the point $\boldsymbol{x}$ has zero utility. Assuming a GP is used for emulation, $y(\boldsymbol{x})$ follows a normal distribution whose mean and variance are given in Eqs. (9) and (10), respectively. Using the reparameterization trick (see Appendix A) we can show that the resulting AF based on $I_{PI}(\boldsymbol{x})$ is [41,42]:

$$\alpha_{PI}(\boldsymbol{x}) = \Phi\left(\frac{\mu(\boldsymbol{x}) - y^*}{\sigma(\boldsymbol{x})}\right) \tag{13}$$

where $\mu(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are defined in Eqs. (9) and (10), and $\Phi(z)$ is the cumulative density function (CDF) of the standard normal random variable z. Eq. (13) clearly indicates that $\alpha_{PI}(\boldsymbol{x})$ favors exploitation because $\Phi(z)$ is maximized at locations where the predictions are close to y^* and have small uncertainty.

In contrast to PI which discards the magnitude of improvement (regardless of the magnitude of $y(\boldsymbol{x})$, $I_{PI}(\boldsymbol{x})$ is either 0 or 1), EI rewards large improvements over y^* by adopting the following utility function:

$$I_{EI}(\boldsymbol{x}) = \max(y(\boldsymbol{x}) - y^*, 0) \tag{14}$$

The corresponding AF can now be obtained by substituting Eq. (14) in Eq. (11) and using the reparameterization trick (see Appendix A for the details):

$$\alpha_{EI}(\boldsymbol{x}) = (\mu(\boldsymbol{x}) - y^*)\Phi(\frac{\mu(\boldsymbol{x}) - y^*}{\sigma(\boldsymbol{x})}) + \sigma(\boldsymbol{x})\phi(\frac{\mu(\boldsymbol{x}) - y^*}{\sigma(\boldsymbol{x})}) \tag{15}$$

where $\mu(\boldsymbol{x})$ and $\sigma(\boldsymbol{x})$ are given in Eqs. (9) and (10), respectively, and $\phi(z)$ is the probability density function (PDF) of z. Eq. (15) clearly demonstrates that $\alpha_{EI}(\boldsymbol{x})$ strikes a balance between exploration and exploitation when it is used as the objective function of the auxiliary optimization problem in Algorithm 1: while the second term on the right-hand side directly deals with uncertainty and hence encourages exploration, the first term favors exploitation [43,44].

Another widely used AF is KG which, unlike PI and EI, is look-ahead because it chooses $\boldsymbol{x}^{k+1}$ (see Algorithm 1) based on the effect of the yet-to-be-seen observation (i.e., y^{k+1} which follows a normal distribution) on the optimum value predicted by the emulator. Following the terminology and setup of Algorithm 1, this AF quantifies the expected utility of $\boldsymbol{x}$ at iteration $k+1$ as:

$$\alpha_{KG}(\boldsymbol{x}) = \mathbb{E}_{p(y|\boldsymbol{x},\mathcal{D}^k)}[\max \mu^{k+1}(\boldsymbol{x})] - \max \mu^k(\boldsymbol{x}) \tag{16}$$

where $\max \mu^k(\boldsymbol{x})$ denotes the maximum mean prediction of the GP trained on $\mathcal{D}^k$. The expectation operation in Eq. (16) appears due to the fact that y^{k+1} is not observed yet and $\alpha_{KG}(\boldsymbol{x})$ is relying on the predictive distribution provided by the GP that is trained on $\mathcal{D}^k$. This expectation cannot be calculated analytically and hence a Monte Carlo estimate is used in practice:

$$\alpha_{KG}(\boldsymbol{x}) \approx \frac{1}{M} \sum_{m=1}^{M} \max \mu^{k+1^m}(\boldsymbol{x}) - \max \mu^k(\boldsymbol{x}) \tag{17}$$

where $\max \mu^{k+1^m}(\boldsymbol{x})$ is calculated by first drawing a sample at $\boldsymbol{x}$ from the GP that is trained on $\mathcal{D}^k$ and then retraining the GP on $\mathcal{D}^k \cup (\boldsymbol{x}, y^m)$ where y^m is response of the drawn sample. In practice, a small value must be chosen for M since maximizing $\alpha_{KG}(\boldsymbol{x})$ over the input space at each iteration of BO is very expensive. We refer the readers to [45,46] for more information on KG and its implementation.

2.2.2. Existing multi-fidelity BO techniques

As stated in Section 1, the overall computational efficiency of BO can be increased by leveraging inexpensive LF datasets. MF BO has been successfully used in many applications such as hyperparameter tuning [47–50], finding Pareto fronts in multi-objective optimizations [51–53], and solving non-linear state-space models [54,55]. For MFBO, both the emulator and the AF must accommodate the multi-source and unbalanced[7] nature of the data.

Co-Kriging, which is an extension of Kriging (or GP), is a popular emulator that handles MF data by reformulating the covariance function in Eq. (2) as follows (assuming there are three data sources denoted by A, B, and C):

$$\text{cov}([y_A(\boldsymbol{x}), y_B(\boldsymbol{x}), y_C(\boldsymbol{x})]^T, [y_A(\boldsymbol{x}'), y_B(\boldsymbol{x}'), y_C(\boldsymbol{x}')]^T) = \Sigma \otimes r\left(\boldsymbol{x}, \boldsymbol{x}'\right) \tag{18}$$

where $\otimes$ denotes the Kronecker product and Σ is a symmetric positive-definite matrix of size 3×3. This reformulation assumes that all the responses (regardless of the source) follow a multi-variate normal distribution and that the matrix Σ_{ij} captures the overall correlation between sources i and j.[8] While this method can fuse any number of data sources it fails to accurately capture cross-source correlations since the matrix Σ has insufficient learning capacity.

Another well-known MF emulation method is that of Kennedy and O'Hagan who fuse bi-fidelity datasets by learning a discrepancy function that aims to explain the differences between HF and LF sources. While this bi-fidelity emulator has proved useful in a wide range of applications [57,58], it has some major drawbacks such as the inability to jointly learn from more than two sources, numerical issues, and assuming a priori additive relation between the discrepancy function and the two data sources.

The bi-fidelity approach of Kennedy and O'Hagan can be viewed as a special case of hierarchical MF modeling where it is assumed that the relative accuracy between all the data sources is known. Space mapping techniques belong to this category, but they are rarely used for sequential sampling, BO, or MF modeling (see [59] for a bi-fidelity example). These techniques are typically employed in solving partial differential equations, particularly to accelerate the convergence of an HF simulation (e.g., based on fine discretization) by initializing it via the results of an LF simulation.

Other notable works are MF polynomial chaos Kriging (MF-PCK) [60] and that of Chen et al. [61]. While both of these works accommodate multiple sources that are non-hierarchically ordered, they presume similar assumptions to KOH (e.g., using an additive bias), have high computational cost, and are very sensitive to the presumed priors.

Upon reformulating the covariance function in Eq. (2), GPs can also be used for hierarchical MF modeling. For instance, the single-task MF GP of the popular BoTorch package adopts an additive covariance function that relies on introducing two user-defined quantitative features [24,28]. The first feature, denoted by x_a, is restricted to the $[0, 1]$ range and assigns a fidelity value to a source based on the user's belief (larger values correspond to higher fidelities). This assigned fidelity value directly affects the correlation and cost function. The second feature, denoted by x_b, is the iteration fidelity parameter and benefits MF BO specifically in the context of hyperparameter tuning of large machine learning models. The covariance function directly uses these two additional features as follows [62]:

$$\text{cov}(\boldsymbol{x}, \boldsymbol{x}') = c_0(\boldsymbol{x}, \boldsymbol{x}') + e_1(x_a, x_a')c_1(\boldsymbol{x}, \boldsymbol{x}') + e_2([x_a, x_b]^T, [x_a', x_b']^T)c_2(\boldsymbol{x}, \boldsymbol{x}') + e_3(x_b, x_b')c_3(\boldsymbol{x}, \boldsymbol{x}') \tag{19}$$

[7] LF sources contribute more samples to the training data since they are typically much cheaper to query from.

[8] GPs can handle multi-response datasets in a similar manner, see [56].

where $c_i(\boldsymbol{x}, \boldsymbol{x}')$ are Matern kernels[9] that characterize the spatial correlations across the numerical inputs and $e_i(\cdot)$ are user-defined functions that model the cross-source correlations. $e_1(x_a, x'_a)$ and $e_3(x_b, x'_b)$ are bias kernels that aim to take the discrepancies among the sources into account while $e_2([x_a, x_b]^T, [x'_a, x'_b]^T)$ models the fidelities' interaction kernel (see Appendix B for more details).

Despite being useful, Eq. (19) has some limitations. For instance, it requires a priori knowledge about the exact hierarchy of fidelities and how they should be encoded as a numerical feature (i.e., x_a). Additionally, the manually-defined functions $e_i(\cdot)$ are insufficiently flexible to learn complex cross-source relations and they also do not provide any learned metric that quantifies which LF sources are useful for MF BO.

Compared to the few emulators described above, the diversity of the AFs in MF BO is more since they are often tailored to the application, see [28,29,63,64]. Many of these developments leverage existing AFs that are used in SF BO such as EI, PI, upper confidence bound [65], Thompson sampling [66], or GP-predictive entropy search [67]. One specific example is most likely EI (MLEI) [68] which is tailored to direct policy search problems where the EI in Eq. (15) is first scaled by multiple context-specific priors and then the resulting AFs are optimized to determine the next candidate point. As another example, Wu et al. [50] develop trace-aware KG to accelerate the hyperparameter tuning process of machine learning models whose training relies on minimizing the loss function (defined as the expected prediction error on the validation data). MF BO is useful in this process since the evaluation accuracy of the loss function can be controlled by parameters such as the number of iterations and training/validation data points. Correspondingly, trace-aware KG adjusts these parameters to use LF but inexpensive evaluations of the loss function (and it trace) during training. We highlight that EI is also widely used in hyperparameter tuning problems [69–71].

3. Proposed approach

As was previously stated, the emulator and AF are the two fundamental components of any BO framework. In this Section, we first discuss the rationale for using LMGP as the emulator of our MF BO framework in Section 3.1 and then introduce our novel cost-aware AF in Section 3.2. We elaborate on the convergence conditions and provide an algorithmic summary of our framework in Section 3.3.

3.1. Multi-fidelity emulation via LMGP

As schematically illustrated in Fig. 2, MF emulation via LMGPs is quite straightforward [32]: Assuming there are ds data sources, we augment the inputs with the additional *categorical* variable $s = \{'1', \ldots, 'ds'\}$[10] whose jth element corresponds to data source j for $j = 1, \ldots, ds$. After this augmentation, the inputs and outputs of all the datasets are concatenated as (following the notation of Fig. 2):

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}_1 & '\mathbf{1}'_{n_1\times 1} \\ \boldsymbol{U}_2 & '\mathbf{2}'_{n_2\times 1} \\ \vdots & \vdots \\ \boldsymbol{U}_{ds} & '\mathbf{ds}'_{n_{ds}\times 1} \end{bmatrix} \text{ and } \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_{ds} \end{bmatrix} \tag{20}$$

where the subscripts $1, 2, \ldots, ds$ correspond to the data sources, n_j is the number of samples obtained from $s(j)$ (i.e., source j), $\boldsymbol{U}_j$ and $\boldsymbol{y}_j$ are, respectively, the $n_j \times (dx + dt)$ feature matrix and the $n_j \times 1$ vector of responses obtained from $s(j)$, and $'\boldsymbol{j}'$ is a categorical vector of size $n_j \times 1$ whose elements are all set to $'j'$. Once the $\{\boldsymbol{U}, \boldsymbol{y}\}$ dataset is built, it is directly fed into LMGP to build an MF emulator.

We argue that, compared to the existing techniques (see Section 2.2.2), LMGPs provide a more flexible and accurate mechanism to build MF emulators, see Fig. 3 for a comparison study on an analytic example. This superiority is because LMGP learns the relations between the sources (which are hidden in the combined datasets) in a manifold. This manifold is learned nonlinearly by embedding the learned latent variables in an exponential function (Eq. (4)) and hence has a much higher representation power than methods that rely on linear operations, e.g., the matrix Σ in co-Kriging that linearly scales the correlations, see Eq. (18). Similarly, LMGP has major advantages over single-task GPs (STGPs) reviewed in Section 2.2.2 because (1) it does not assume any hierarchy across the

[9] The parameters of these kernels are endowed with Gamma priors in BoTorch.

[10] We use quotation to indicate that the elements of s are categorical, e.g., $'1'$ is not a quantitative number.

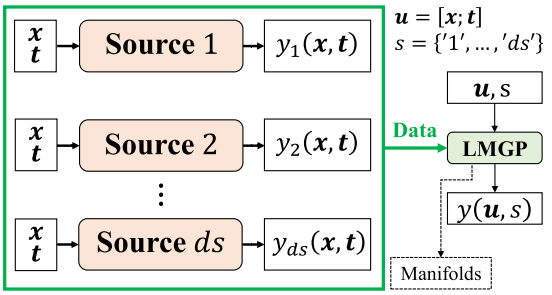


Fig. 2. Multi-fidelity emulation with LMGP: The training data is built by first augmenting the inputs with the *categorical* feature s that denotes the data source of a sample and then concatenating all the inputs and outputs, see Eq. (20). LMGP can use one or more manifolds to encode the categorical variables into a quantitative space. For MF emulation, we recommend using two manifolds to simplify the visualization of cross-source relations: one manifold for s and the other for the rest of the categorical variables, i.e., $\boldsymbol{t}$.

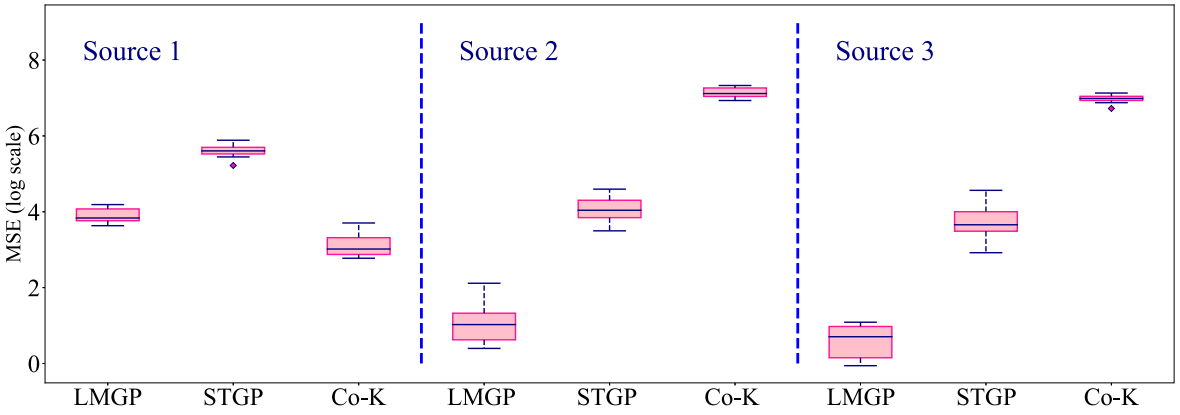


Fig. 3. Emulator comparison: We compare the prediction accuracy of LMGP against single-task multi-fidelity GP (STGP) and Co-Kriging (Co-K) on an analytic problem with three sources (see Borehole in Table 1 where HF, LF3, and LF4 are used). While LMGP and STGP use $(5, 5, 50)$ initial data for (HF, LF3, LF4) sources, Co-K uses $(50, 50, 50)$ to achieve comparable performance. Each emulator is trained 10 times by randomizing the initial data. It is evident that LMGP consistently outperforms other methods in emulating all the sources (including the effect of random initialization). Prediction accuracy is measured by calculating the mean squared error (MSE) on unseen test data. The large variations in the MSEs of LMGP are due to the *log* scale representation, see Appendix C.

data sources, (2) the cross-source relations are encoded via learned latent variables which have significantly higher representation power than a single user-defined scalar variable (see Eq. (19)) that directly affects the covariance function of the underlying GP and requires knowledge of the relative source fidelities.

As shown in Fig. 3, LMGPs build MF emulators that more accurately learn all the sources (rather than just the HF source). While we can alter the formulations in Section 2.1 such that LMGPs prioritize learning the HF source, we do not believe this is a good general decision in the context of MF BO. The reasoning behind our belief is that the quality of LF predictions (obtained via the emulator) greatly affects the exploration nature of BO and, hence, its convergence behavior. The empirical results in Section 4 strongly support this reasoning.

Another major advantage of LMGPs is that their learned manifold provides an intuitive and visualizable global metric for comparing the relative discrepancies/similarities among the data sources, see Fig. 6 for four examples with different number of data sources. This manifold is particularly useful in detecting anomalous sources whose data adversely affects MF BO. We demonstrate this in Section 4 with some examples where we use the initial MF data (i.e., before starting the BO iterations) to correctly predict whether SF BO outperforms MF BO.

Given the importance of identifying relative discrepancies among data sources, we slightly modify the correlation function of LMGP to learn two manifolds where the first one encodes the original categorical variables (collectively denoted by $\boldsymbol{t}$ in Section 2.1) while the second one encodes the data source identifier (denoted by s in Fig. 2). The

new correlation function is (compare to Eq. (5)):

$$r(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{t}\\ s\end{bmatrix}, \begin{bmatrix}\boldsymbol{x}'\\ \boldsymbol{t}'\\ s'\end{bmatrix}) = \exp\{-\sum_{i=1}^{dx} 10^{\omega_i}(x_i - x_i')^2 \ - \sum_{i=1}^{dz}(z_i(\boldsymbol{t}) - z_i(\boldsymbol{t}'))^2 - \sum_{i=1}^{dh}(h_i(s) - h_i(s'))^2\} \tag{21}$$

where $\boldsymbol{h}(s) = [h_1, \ldots, h_{dh}]^T$ is the latent representation of data source s. Similar to Eq. (6), this latent representation is obtained by post-multiplying a prior vector by the parametric matrix $\boldsymbol{A}_h$[11]:

$$\boldsymbol{h}(s) = \boldsymbol{\zeta}(s)\boldsymbol{A}_h \tag{22}$$

In our studies, we always design the prior by one-hot-encoding the categorical variable $s = \{'1', \ldots, 'ds'\}$ that identifies the data source and estimate all of LMGP's hyperparameters via MAP with the new $\boldsymbol{R}$ built using Eq. (21).

$$[\widehat{\boldsymbol{\beta}}, \hat{\sigma}, \widehat{\boldsymbol{\omega}}, \widehat{\boldsymbol{A}}, \widehat{\boldsymbol{A}}_h] = \underset{\boldsymbol{\beta},\sigma^2,\omega,\boldsymbol{A},\widehat{\boldsymbol{A}}_h}{\operatorname{argmax}} |2\pi\sigma^2\boldsymbol{R}|^{-\frac{1}{2}} \times \exp\{\frac{-1}{2}(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\beta})^T(\sigma^2\boldsymbol{R})^{-1}(\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\beta})\} \times \underset{\boldsymbol{\beta},\sigma^2,\omega,\boldsymbol{A},\widehat{\boldsymbol{A}}_h}{\mathrm{P}(\cdot)} \tag{23}$$

Then, We use Eq. (21) to explain the relation between the latent fidelity representations, i.e., $\boldsymbol{h}(s)$, and the relative fidelity of the data sources. At the same inputs, the correlation between the estimated outputs of sources s and s' is:

$$0 \leq corr(y_s(\boldsymbol{x}, \boldsymbol{t}), y_{s'}(\boldsymbol{x}, \boldsymbol{t})) = r(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{t}\\ s\end{bmatrix}, \begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{t}\\ s'\end{bmatrix}) = \exp\{0 - 0 - \sum_{i=1}^{d_h}(h_i(s) - h_i(s'))^2\} \leq 1 \tag{24}$$

So sources with similar fidelities which provide highly correlated responses, must have similar latent representations, i.e., $\boldsymbol{h}_i(s) \simeq \boldsymbol{h}_i(s')$. This relation is illustrated in Fig. 6 where sources with similar fidelities are encoded by close-by points in the manifold.

3.2. Multi-source cost-aware acquisition function

The choice of AF affects the performance of BO quite significantly. This choice is especially important in MF BO because, in addition to balancing exploration and exploitation, the AF has to consider the biases of LF data and source-dependent sampling costs. To demonstrate these challenges, consider the analytic example in Fig. 4(a) where, while the two functions are correlated, the LF source's global optimum (the location and the corresponding y value) is quite different than that of the HF source. Since LF sources are typically much cheaper than the HF source, a naive AF (that merely scales the expected utility based on the cost) forces MF BO to converge to the global optimum of the LF source, see Fig. 4(b) for a one-dimensional example.

Contrary to existing approaches, we argue that the key to addressing the above challenges is to quantify the information value of LF and HF data based on different metrics which are then compared against each other to determine the candidate input and the corresponding source. In particular, we propose to use the LF sources exclusively for exploration to leverage their correlations with the HF source while preventing them from dominating the convergence behavior of MF BO. Additionally, we propose to exclusively employ the HF source for exploitation to maximally use its trustworthy samples[12] during optimization.

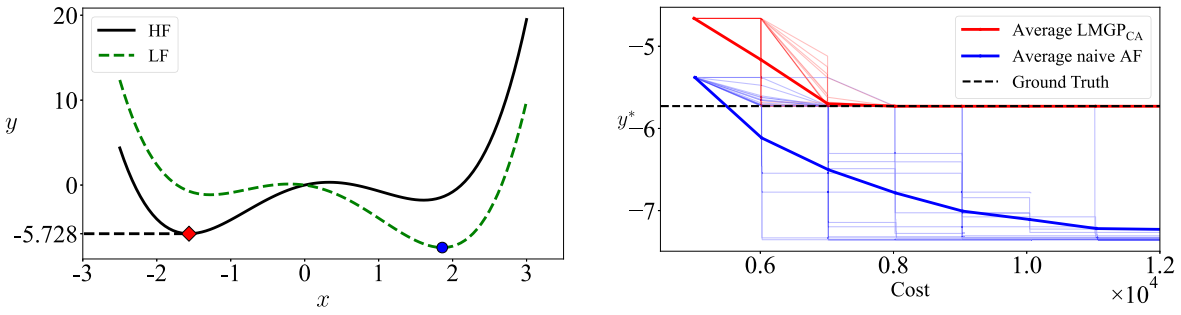
To develop the AF for the jth LF source with $j = [1, \ldots, ds]$ and $j \neq l$ where l denotes the HF source, we follow Section 2.2.1 and define the improvement (for a maximization problem) as $y_j(\boldsymbol{x}) - y_j^*$ where $y_j(\boldsymbol{x})$ denotes the LMGP-based prediction at $\boldsymbol{x}$ for source j and y_j^* is the best function value in the obtained dataset from source j. We use the reparameterization trick to rewrite this improvement as:

$$\frac{y_j(\boldsymbol{x}) - \mu_j(\boldsymbol{x})}{\sigma_j(\boldsymbol{x})} = z \sim \mathcal{N}(0, 1) \Rightarrow y_j(\boldsymbol{x}) - y_j^* = (\mu_j(\boldsymbol{x}) - y_j^*) + \sigma_j(\boldsymbol{x})z \tag{25}$$

where $\mu_j(\boldsymbol{x})$ and $\sigma_j(\boldsymbol{x})$ are defined in Eqs. (9) and (10), respectively. In Eq. (25) the $(\mu_j(\boldsymbol{x}) - y_j^*)$ and $\sigma_j(\boldsymbol{x})z$ terms control the exploitation and exploration aspects of the improvement, respectively. We now define our utility function

[11] $\boldsymbol{A}_h \sim \mathcal{N}(0, 3)$.

[12] These samples may be corrupted via $\epsilon \sim \mathcal{N}(0, \sigma^2)$ where σ^2 is the (unknown) noise variance.



(a) Biased LF source **(b) Effect of AF on convergence**

Fig. 4. Double-well potential: The LF source has a systematic bias because its optima do not match with those of the HF source. However, since the two curves have similar trends, an effective AF can leverage this correlation to reduce the overall data acquisition costs. The LF and HF sources have a cost-per-sample of, respectively, 1 and 1000 in this example. To assess the stability of the results, this example is run 20 times where the thin curves demonstrate the convergence of each repetition and the thick one is their average. With the naive AF, expected utility is divided by the sampling cost and hence MF BO primarily queries the cheap LF source. Lack of HF samples and a mechanism that ensure the found optimum belongs to the HF source result in convergence to a wrong solution.

that focuses on exploration by dropping the first term on the far right-hand-side of Eq. (25):

$$I_{LF}(\boldsymbol{x};\, j) = \begin{cases} \sigma_j(\boldsymbol{x})z & y_j(\boldsymbol{x}) > y_j^* \\ 0 & y_j(\boldsymbol{x}) \leq y_j^* \end{cases} \tag{26}$$

which is used for the jth LF source in our framework. We obtain the corresponding AF by substituting $I_{LF}(\boldsymbol{x};\, j)$ in Eq. (11):

$$\alpha_{LF}(\boldsymbol{x};\, j) = \int_{-\infty}^{\infty} I_{LF}(\boldsymbol{x};\, j)\phi(z)\mathrm{d}z = \int_{-\infty}^{\infty} \sigma_j(\boldsymbol{x})z\phi(z)\mathrm{d}z \tag{27}$$

The integral is zero for $y_j(\boldsymbol{x}) < y_j^*$ so we find the corresponding switch point in terms of z:

$$y_j(\boldsymbol{x}) = y_j^* \Rightarrow \mu_j(\boldsymbol{x}) + \sigma_j(\boldsymbol{x})z = y_j^* \Rightarrow z_0 = \frac{y_j^* - \mu_j(\boldsymbol{x})}{\sigma_j(\boldsymbol{x})} \tag{28}$$

Inserting Eq. (28) in Eq. (27) yields:

$$\begin{aligned} \alpha_{LF}(\boldsymbol{x};\, j) &= \int_{z_0}^{\infty} \sigma_j(\boldsymbol{x})z\phi(z)\mathrm{d}z = \int_{z_0}^{\infty} \frac{\sigma_j(\boldsymbol{x})z}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}\mathrm{d}z \\ &= \frac{\sigma_j(\boldsymbol{x})}{\sqrt{2\pi}} \int_{z_0}^{\infty} z e^{-\frac{z^2}{2}}\mathrm{d}z = \frac{\sigma_j(\boldsymbol{x})}{\sqrt{2\pi}} \int_{z_0}^{\infty} (e^{-\frac{z^2}{2}})'\mathrm{d}z \\ &= -\frac{\sigma_j(\boldsymbol{x})}{\sqrt{2\pi}} [e^{-\frac{z^2}{2}}]_{z_0}^{\infty} = \sigma_j(\boldsymbol{x})\phi(z_0) = \sigma_j(\boldsymbol{x})\phi(\frac{y_j^* - \mu_j(\boldsymbol{x})}{\sigma_j(\boldsymbol{x})}) \end{aligned} \tag{29}$$

Comparison between this AF and Eq. (15) illustrates that our proposed AF for LF sources is the same as the exploration part of EI.

For the HF source, we propose to use PI as the AF because it focuses on exploitation and is computationally efficient (the efficiency is due to the analytic form of PI which dispenses with expensive numerical integration). Assuming source l provides the HF data, this AF is given by:

$$\alpha_{HF}(\boldsymbol{x}; l) = \Phi(\frac{\mu_l(\boldsymbol{x}) - y_l^*}{\sigma_l(\boldsymbol{x})}) \tag{30}$$

We use $\alpha_{HF}(\boldsymbol{x}; l)$ and $\alpha_{LF}(\boldsymbol{x};\, j)$ as defined above in each iteration of BO to solve ds auxiliary optimization problems (assuming there are ds data sources) to find the candidate location with the highest expected utility from each source. We then find the final candidate point and the corresponding source by comparing the cost-normalized

version of these ds values. Hence, assuming source l provides HF data among the ds sources, our AF that considers source-dependent costs and fidelities is:

$$\alpha_{MFCA}(\boldsymbol{x}; j) = \begin{cases} {}^{\alpha_{LF}(\boldsymbol{x};j)}/_{O(j)} & j = [1, \ldots, ds] \ \ \& \ \ j \neq l \\ {}^{\alpha_{HF}(\boldsymbol{x};l)}/_{O(l)} & j = l \end{cases} \tag{31}$$

and

$$[\boldsymbol{x}^{k+1}, j^{k+1}] = \underset{\boldsymbol{x}, j}{\text{argmax}} \ \ \alpha_{MFCA}(\boldsymbol{x}; j) \tag{32}$$

where $O(j)$ is the cost associated with taking a single sample from source j. In practice, $O(j)$ can be the cost of an experiment or simulation with units in dollars, CPU/GPU time, or any other factor that an analyst aims to consider during BO (e.g., complexity, effort) . $\boldsymbol{x}^{k+1}$ is the point that source $j^{k+1} \in \{1, \ldots, ds\}$ must be evaluated in the current BO iteration. We now point out several important aspects of Eq. (31).

The naive AFs in Eqs. (29) and (30) quantify the value of a sample by comparing it to the best available sample for the corresponding source (and not the best sample across all the sources). The advantage of this source-wise comparison in each of the AFs is that it encourages sampling from sources that provide larger values (which is desirable for a maximization problem). However, this formulation enables LF sources whose optima are larger than the HF source to dominate the optimization process where not only more samples are taken from these LF sources (which may also cause numerical issues), but also the converged solution does not belong to the profile of the HF source. This issue is exacerbated once the AFs are divided by the data collection costs (see Eq. (31)) since LF samples are (typically) much cheaper than HF data.

The abovementioned issues are addressed with three mechanisms in our MFCA approach. Firstly, we always report the points sampled from the HF source as the final optimization history (this choice ensures that the final solution indeed belongs to the profile of the HF source but it does not guarantee global optimality[13]). Secondly, we always use the fidelity manifold of the LMGP that is trained on the *initial* data to detect the LF sources that should not be used in BO due to their severe discrepancy (we demonstrate the benefit of this exclusion with examples in Section 4). Thirdly, we have designed the core of our AFs for the HF and LF sources based on, respectively, the CDF and the scaled PDF of the standard normal variable z. As detailed below and empirically shown in Section 4, the intricate relation between these two functions during the optimization reduces the effect of LF sources' biases on the convergence.

As illustrated in Fig. 5, $\phi(z)$ and $\Phi(z)$ have comparable values up to $\mathbb{E}[z] = 0$ but the ratio $\Phi(z)/\phi(z)$ increasingly grows as z realizes larger values. This trend indicates that if an HF candidate point sufficiently improves y_l^*, then Eq. (32) queries the HF source at that point to obtain a new sample for the next BO iteration. The frequency of this query during the optimization process is controlled by the data collection cost and $\sigma_j(\boldsymbol{x})$, see Eqs. (29) and (30), respectively. If HF samples are highly costly, the auxiliary optimization in Eq. (32) reduces the sampling frequency. However, unlike existing approaches such as BoTorch, this reduction does not translate into "never sampling from the HF source" (even if the cost ratios are as large as 1000, see Section 4) because $\Phi(z)/\phi(z)$ can be quite large. Regarding the $\sigma_j(\boldsymbol{x})$ term in Eq. (29), we note that it encourages exploring the regions where LMGP provides highly uncertain predictions for an LF source. This scenario happens when an LF source is rarely sampled and there are insufficient correlations between that source and other sources.

In summary, our proposed AF in Eq. (31), while involving intricate interactions between the fidelities and costs, has a simple form which is analytic (and hence computationally efficient) and interpretable. As we illustrate in Section 4 this AF, combined with LMGP, dramatically improves the performance of our MFCA BO framework.

3.3. Convergence metric

Similar to AFs, convergence criteria of MF BO techniques are traditionally tailored to the application since many factors (e.g., budget constraints, numerical issues, or convergence history) affect the results. We believe the emulator and AF of our framework alleviate many of the convergence issues associated with MF BO and hence use two simple convergence metrics: overall costs and maximum number of iterations without improvement. The former is a rather generic metric but it can result in considerably high number of iterations if one of the LF sources

[13] BO does not guarantee global optimality anyways.

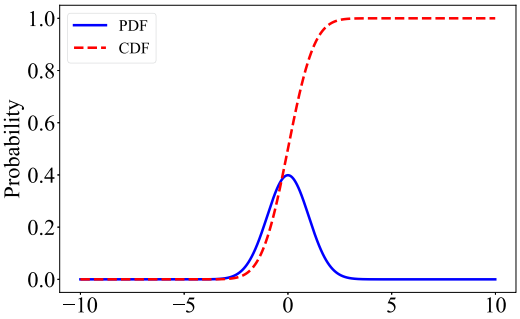


Fig. 5. Standard normal variable: The PDF and CDF of $z \sim \mathcal{N}(0, 1)$ have comparable magnitudes up to the mean of z (i.e., 0) but increasingly differ after 0.

Algorithm 2: Multi-fidelity Cost-aware Bayesian Optimization for Maximization

Given: Initial multi-fidelity data $\mathcal{D}^k = \{(\boldsymbol{x}^i, y^i\}_{i=1}^k$, black-box functions $f(\boldsymbol{x}; j)$ and their corresponding sampling costs $O(j)$ where $j = [1, ..., ds]$
Goal: Optimizing high-fidelity function (source $l \in [1, ..., ds]$)
Define: Utility functions (see Eqs. (12) and (26)) and stop conditions
Step 0**:** Train an `LMGP` and exclude highly biased low-fidelity sources based on its fidelity manifold
while *stop conditions not met* **do**

1. Train an `LMGP` using $\mathcal{D}^k$
2. Define the multi-fidelity cost-aware acquisition function (see Eqs. (29), (30), and (31)):
$$\alpha_{MFCA}(\boldsymbol{x}; j) = \begin{cases} \alpha_{LF}(\boldsymbol{x}; j)/O(j) & j \in \{1, \cdots, ds\} \quad \& \quad j \neq l \\ \alpha_{HF}(\boldsymbol{x}; l)/O(l) & j = l \end{cases}$$
3. Solve the auxiliary optimization problem:
$[\boldsymbol{x}^{k+1}, j^{k+1}] = \underset{\boldsymbol{x}, j}{\text{argmax}} \ \alpha_{MFCA}(\boldsymbol{x}; j)$
4. Query $f(\boldsymbol{x}; j)$ at point $\boldsymbol{x}^{k+1}$ from source j^{k+1} to obtain y^{k+1}
5. Update data: $\mathcal{D}^{k+1} \leftarrow \mathcal{D}^k \cup (\boldsymbol{x}^{k+1}, y^{k+1})$
6. Update counter: $k \leftarrow k + 1$

end
Output: Updated data $\mathcal{D}^k = \{(\boldsymbol{x}^i, y^i)\}_{i=1}^k$, `LMGP`

is extremely inexpensive to query. For this reason, we recommend using additional metrics (such as the second one above) that track convergence.

Algorithm 2 summarizes our framework for MFCA BO. Compared to Algorithm 1, the major differences are in the choice of the emulator and AF which now can handle multi-source data that have different levels of fidelity and cost. In addition, a pre-processing step is added which leverages the fidelity manifold of LMGP to detect the LF sources that must be excluded from the BO process due to their large discrepancies with respect to the HF source.

4. Results and discussions

We compare the performance of the following four methods on four analytic and two real-world examples (Sections 4.1 and 4.2, respectively):

- $\texttt{LMGP}_{\texttt{CA}}$: Our proposed MFCA BO approach.
- $\texttt{LMGP}_{\texttt{EI}}$: Single-fidelity BO whose emulator and AF are LMGP and EI, respectively.
- $\texttt{LMGP}_{\texttt{PI}}$: Single-fidelity BO whose emulator and AF are LMGP and PI, respectively.
- `BoTorch`: Multi-fidelity BO with BoTorch where STGP and KG are used as emulator and AF.

The first three methods are myopic and hence computationally much faster than BoTorch which is lookahead (see Section 2.2.2 for details on BoTorch). Assuming the computational costs of BO (which are mainly associated with emulation and solving the auxiliary optimization problem) are negligible compared to querying any of the data sources, we compare the above methods in terms of cost and accuracy which are defined as, respectively, the overall data collection costs and the ability to find the global optimum of the HF source. In Appendix D we study the costs of the auxiliary optimizations in $\texttt{LMGP}_{\texttt{CA}}$ and BoTorch to illustrate that our approach also performs better in this regard.

Our rationales for comparing $\texttt{LMGP}_{\texttt{CA}}$ against the above three methods are to demonstrate: (1) the advantages of employing LF sources in BO, (2) that our designed myopic AF (see Eq. (31)) improves both the sampling cost and accuracy compared to even lookahead methods such as BoTorch, and (3) the importance of excluding highly biased LF sources from BO. We also note that the emulators in BoTorch cannot handle categorical inputs and hence we compare $\texttt{LMGP}_{\texttt{CA}}$ to $\texttt{LMGP}_{\texttt{EI}}$ and $\texttt{LMGP}_{\texttt{PI}}$ in Section 4.2 where the two examples have categorical variables.

For all the methods, we terminate the optimization process if either of the following conditions are met: (1) the overall sampling cost reaches a pre-determined maximum budget, or (2) the best HF sample (i.e., y_l^* in Eq. (30)) does not improve over 50 consecutive iterations. These conditions are quite simple and straightforward; allowing us to focus on the effects of AF and the emulator on the performance.

In Section 4.1, we set the maximum budget to 40 000 units for $\texttt{LMGP}_{\texttt{CA}}$, $\texttt{LMGP}_{\texttt{EI}}$, and $\texttt{LMGP}_{\texttt{PI}}$ to ensure that there are enough iterations that the competing approaches converge (especially in high-dimensional examples). However, we do not choose a very large budget (e.g., 100 000) to avoid very long run-times. For BoTorch we choose 50 000 since it is, as demonstrated below, highly inefficient and requires more samples to provide reasonable accuracy. In all the examples of Section 4.1 an HF sample costs 1000 so $\texttt{LMGP}_{\texttt{EI}}$ and $\texttt{LMGP}_{\texttt{PI}}$ are terminated based on the maximum budget condition. In Section 4.2, we set the maximum budget to 1000 and 1800 for the two examples since their data collection cost are much lower than the examples in Section 4.1.

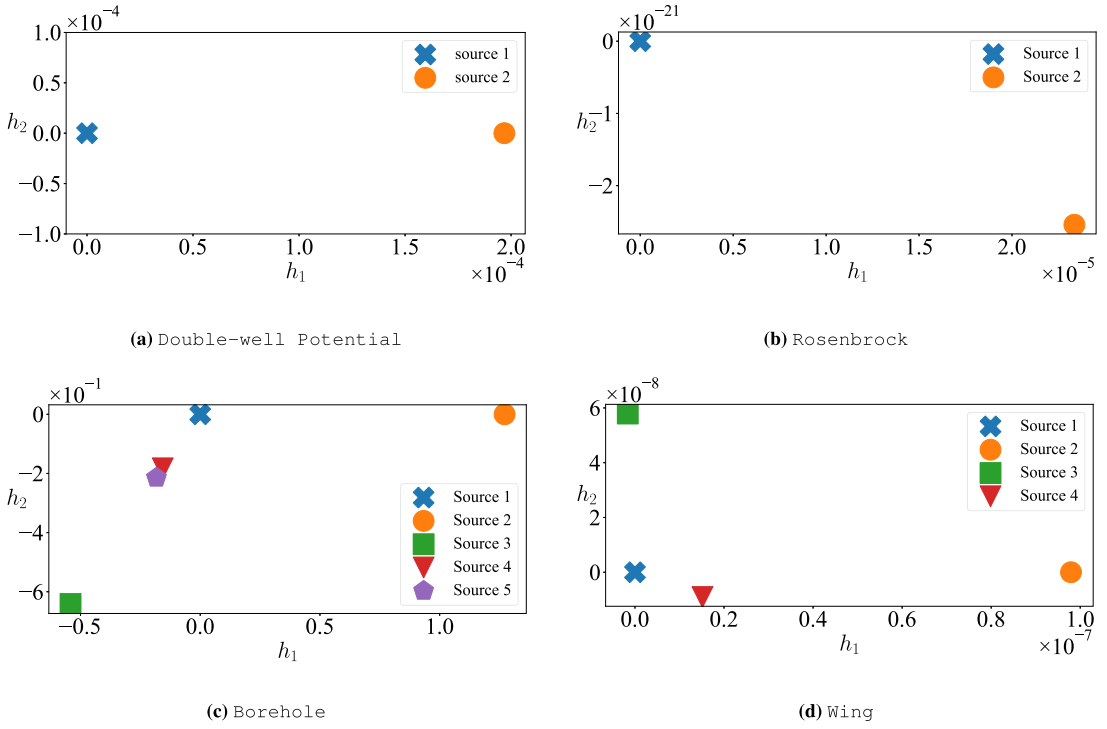
4.1. Analytic examples

As detailed in Table 1 in Appendix E, we consider four analytic examples (Double-well Potential, Rosenbrock, Borehole, and Wing) whose input dimensionality ranges from 1 to 10. All examples are single-response and the number of data sources varies between 2 and 5. The source-dependent sampling costs and number of initial data points are also detailed in Table 1. To compare the robustness and effectiveness of the four BO methods described above, we use relatively small initial datasets (especially from the HF source) and consider various cost ratios (the maximum cost ratio between two sources equals 1000). For each example, we quantify the effect of random initial data by repeating the optimization process 20 times for each method (all initial data are generated via Sobol sequence). In Appendix F we also study the effect of the sizes of the initial datasets for the Borehole example and demonstrate that the performance of $\texttt{LMGP}_{\texttt{CA}}$ is quite insensitive to them.

Table 1 also enumerates the relative accuracy of the LF sources of each example by calculating the relative root mean squared error (RRMSE) between them and the corresponding HF source based on 1000 samples (these RRMSEs are not used in BO in any way). In the case of Borehole we observe that, unlike Wing, the source ID, true fidelity level (based on the RRMSEs), and sampling cost are not related. For instance, the first LF source is the least accurate and most expensive among all the LF sources in Borehole.

Per Step 0 in Algorithm 2, we always use the initial data to train an LMGP to identify the useful LF sources. Based on Fig. 6, we expect the LF sources to be beneficial in all the examples except for Borehole since the latent points of the first and second sources are distant from the latent position of the HF source (we test this expectation in Appendix G). Hence, hereafter we exclude these highly biased LF sources, i.e., both $\texttt{LMGP}_{\texttt{CA}}$ and BoTorch use three sources (HF, LF3, and LF4 in Table 1) to optimize Borehole. We note that BoTorch does not provide any mechanism to detect highly biased LF sources, but we also leverage this LMGP-based insight to boost the performance of BoTorch and, in turn, better highlight the effectiveness of our proposed AF.

Fig. 7 summarizes the convergence histories by tracking the best HF estimate found by each method (i.e., y_l^* in Eq. (30)) as a function of accumulated sampling costs. It is evident that $\texttt{LMGP}_{\texttt{CA}}$ consistently outperforms all other methods across the four examples. In particular, $\texttt{LMGP}_{\texttt{CA}}$ demonstrates the advantage of leveraging inexpensive LF sources in BO by accelerating the convergence without sacrificing the accuracy (compare $\texttt{LMGP}_{\texttt{CA}}$ to $\texttt{LMGP}_{\texttt{EI}}$ and $\texttt{LMGP}_{\texttt{PI}}$ for any of the examples in Fig. 7). Additionally, unlike BoTorch, the performance of $\texttt{LMGP}_{\texttt{CA}}$ is robust to the



(a) Double-well Potential

(b) Rosenbrock

(c) Borehole

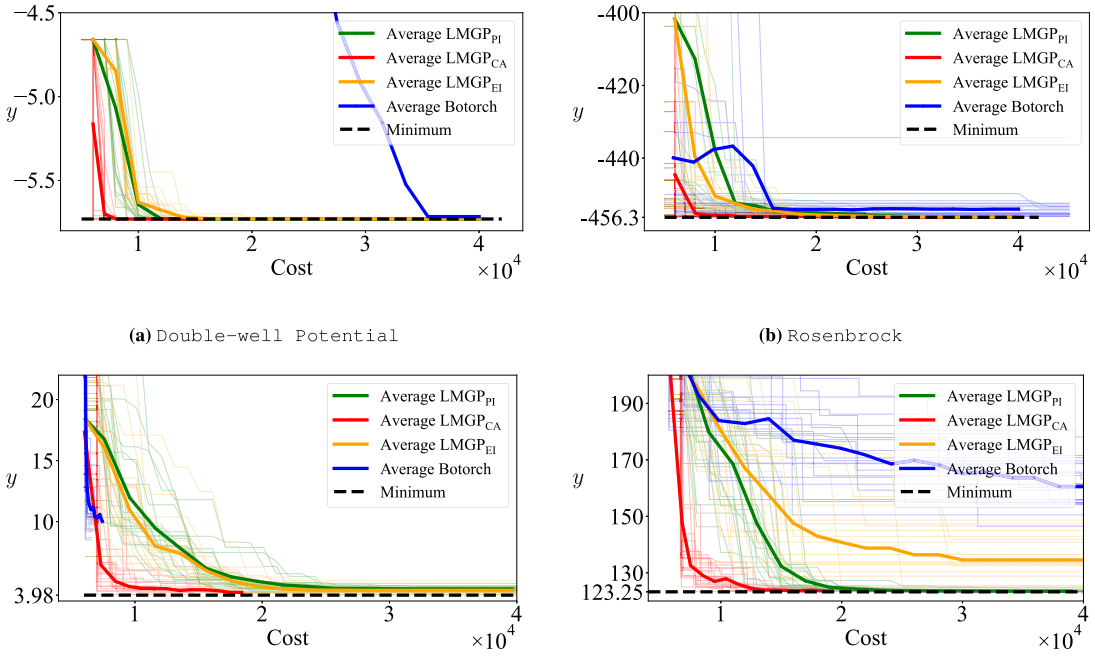
(d) Wing

Fig. 6. Fidelity manifolds of analytic examples: We train LMGPs on the initial multi-source data to (inversely) learn the relation between the sources. The encoding described in Section 3.1 is used and hence each source is represented with a point in the fidelity manifold of LMGP. For each example, the plot corresponds to a randomly selected case among the 20 repetitions (only one plot is shown due to consistency across the repetitions). Based on the distances in the fidelity manifold (see also Eq. (21)), we conclude that the second and third sources are highly biased in `Borehole` and hence must be excluded from the MF BO process.

input dimensionality and sampling costs. For instance, `BoTorch` estimates the optimum as $y_l^* = 10.02$ in Fig. 7(c) while the ground truth is 3.98. The reason behind this inaccuracy is that `BoTorch` fails to find an HF sample whose information value is large enough to overcome its high sampling cost and, as a result, cheap LF sources are largely queried. However, these queries do not improve y_l^* and hence the second strop condition (maximum number of repetitions without improvement of y_l^*) terminates `BoTorch` after 50 iterations. Removing this stop condition, while significantly increasing the number of iterations, does not improve the accuracy of `BoTorch`. To demonstrate the effect of this removal, we only consider the maximum budget constraint for `BoTorch` in `Wing` and observe that the ground truth is again not found, see Fig. 7(d). These issues are resolved in $\texttt{LMGP}_{\texttt{CA}}$ by the intricate interplay between $\Phi(z)$ and $\phi(z)$ as explained in Section 3.2.

To exclude the effect of the termination criteria from the results, we provide the accumulated cost up to and including the iteration at which each method finds its best estimate, see Fig. 8. In terms of finding the true optimum (compare the blue dots to the horizontal dashed line), $\texttt{LMGP}_{\texttt{CA}}$ outperforms all other methods and is followed by $\texttt{LMGP}_{\texttt{EI}}$ and then $\texttt{LMGP}_{\texttt{PI}}$ (the high accuracy of the SF methods in finding the ground truth is expected since they only sample from the HF source and neither of the two termination criteria are stringent). However, `BoTorch` either terminates at an incorrect solution or is even costlier than SF methods. This poor performance is expected since, even though `BoTorch` is lookahead, its AF cannot handle the high cost-ratios across the sources and its emulator does not effectively learn the nonlinear relations between the HF and LF sources.

To provide more insight into the mechanics of MF BO methods, we also report the number of iteration at convergence, see Fig. 9. As expected, `BoTorch` and $\texttt{LMGP}_{\texttt{CA}}$ require more iterations as they aim to leverage cheap LF sources to reduce the overall costs. $\texttt{LMGP}_{\texttt{CA}}$ needs fewer iterations for convergence than `BoTorch` except in Fig. 9(b). This behavior is a result of the termination criteria: the ground truth of `Rosenbrock` is -456.3 while $\texttt{LMGP}_{\texttt{CA}}$ rapidly converges to a very close solution (-456) and then takes highly cheap LF samples to improve the



(a) `Double-well Potential` **(b)** `Rosenbrock`

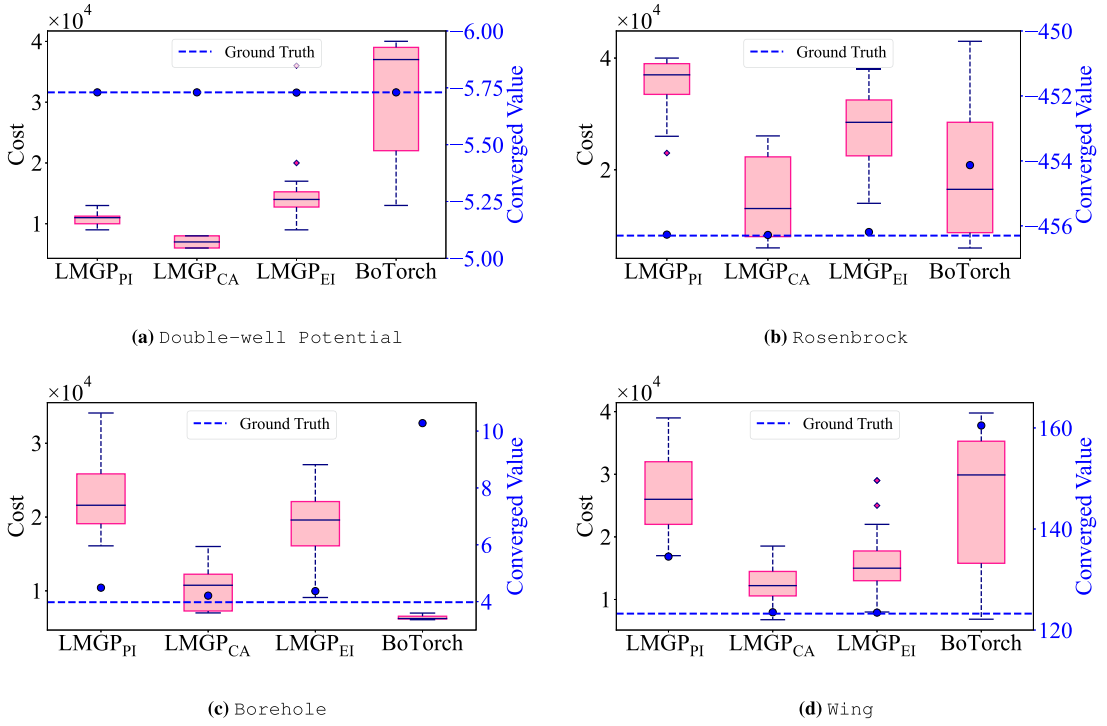
(c) `Borehole` **(d)** `Wing`

Fig. 7. Convergence histories: The plots illustrate the best HF sample (i.e., y_l^* in Eq. (30)) found by each method as a function of sampling costs accumulated during the BO iterations (the cost of initial data is included). $\texttt{LMGP}_{\texttt{CA}}$ consistently outperforms other methods in all the examples, especially in high-dimensional cases (i.e., `Borehole` and `Wing`). The solid thick curves indicate the average behavior across the 20 repetitions (the variations associated with `BoTorch` in `Double-well Potential` are extremely small and hence not visible). $\texttt{LMGP}_{\texttt{CA}}$ and `BoTorch` use three sources in `Borehole`. The second termination condition (i.e., maximum of 50 BO iterations without improvement in y_l^*) is disabled for `BoTorch` in `Wing` to illustrate its convergence trajectory.

best sample. These improvements are quite small (0.01 per iteration) and hence many iterations are needed for convergence.

We highlight that, as long as the associated costs are low, a large number of iterations is not reflective of bad performance since the goal of MF BO is to reduce the overall sampling costs and not necessarily the total number of samples. To demonstrate this, we provide the per-source sampling frequencies for $\texttt{LMGP}_{\texttt{CA}}$ and `BoTorch` in Fig. 10 which demonstrates that $\texttt{LMGP}_{\texttt{CA}}$ automatically adjusts its sampling mechanism based on the initial data and the relative accuracy of the LF sources (compared to the corresponding HF source) and their costs. For instance, unlike `BoTorch` which avoids querying the HF source in `Borehole`, $\texttt{LMGP}_{\texttt{CA}}$ leverages all sources where the number of samples taken from each source depends on its cost and (in the case of LF sources) relative accuracy. In particular, we observe that $\texttt{LMGP}_{\texttt{CA}}$ samples almost equally from the LF sources even though the second source is 10 times cheaper (note that these LF sources correspond to the third and fourth sources in Table 1). This behavior may seem undesirable at the first glance especially since the LF sources have the same relative accuracy (see RRMSEs in Table 1) but a closer look indicates that it is primarily caused by the number of initial samples: since there are 10 times more data points from the second LF source, the emulator of $\texttt{LMGP}_{\texttt{CA}}$ correctly provides large prediction uncertainties which, in turn, results in a large expected utility for the first LF source, see Eq. (29). These discussions also hold for `Wing`, see Fig. 10(d) where, unlike `BoTorch`, $\texttt{LMGP}_{\texttt{CA}}$ samples from the LF sources based on their relative accuracy (which is learnt internally by its emulator) as well as cost.

Finally, we demonstrate the performance of $\texttt{LMGP}_{\texttt{CA}}$ in balancing exploration and exploitation. To this end, source-wise sampling orders made by $\texttt{LMGP}_{\texttt{CA}}$ and `BoTorch` are visualized for a randomly selected repetition in each example, see Fig. 11. As it can be observed, $\texttt{LMGP}_{\texttt{CA}}$ alternates between all the sources: in each example the majority of the samples are from the LF sources which are queried based on exploration (see Eqs. (29) and (31)) while the expensive HF source is typically used with much lower frequency. The sampling orders are particularly



(a) `Double-well Potential` **(b)** `Rosenbrock`

(c) `Borehole` **(d)** `Wing`

Fig. 8. Accumulated costs before improvements plateau: The box-plots illustrate the accumulated costs up to and including the iteration at which the best HF sample is first obtained (i.e., these box-plots do not consider the two termination criteria). On the right axis, the converged solution (averaged across the 20 repetitions) and ground truth are demonstrated via, respectively, the blue marker and the horizontal dashed line. In all four examples, $\texttt{LMGP}_{\texttt{CA}}$ finds the optimum faster than other methods. Comparison between the axes indicates that small accumulated costs do not imply superior performance since the converged solution might be a local optimum, as is the case for `BoTorch` in 8(c). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

interesting for `Rosenbrock` and `Borehole` where, unlike `BoTorch` which struggles to alternate between the sources, $\texttt{LMGP}_{\texttt{CA}}$ effectively uses all sources during the optimization.

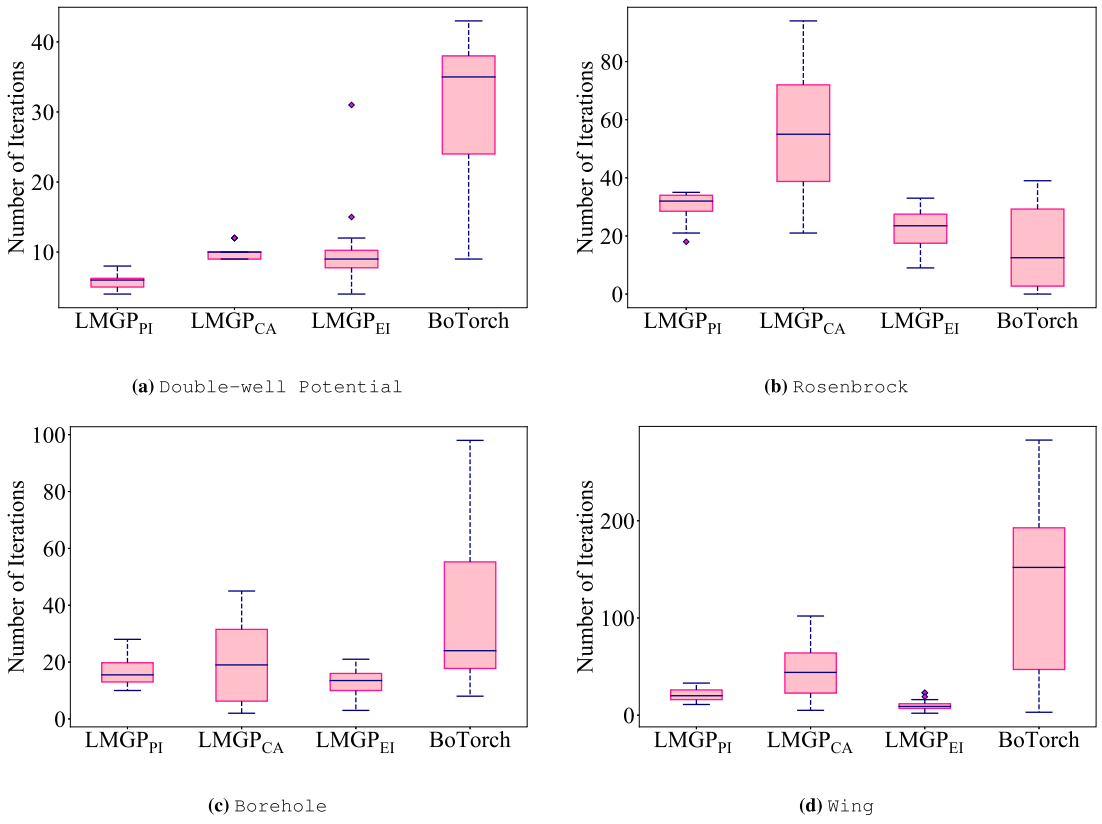
4.2. Real-world datasets

In this Section, we study two materials design problems where the goal is to identify the composition that optimizes the property of interest. Unlike the examples in Section 4.1, these two problems are noisy and have categorical inputs. We compare the performance of $\texttt{LMGP}_{\texttt{CA}}$ only against the SF methods since `BoTorch` does not accommodate categorical inputs. In both examples, the HF and LF data are obtained via simulations (based on the density functional theory) which have different fidelity levels.

The first problem is bi-fidelity and aims to find the member of the nanolaminate ternary alloy (`NTA`) family with the largest bulk modulus [72]. The HF and LF datasets each have 224 samples, one response, and 10 features (7 quantitative and 3 categorical where the latter have 10, 12, and 2 levels). In our studies, we presume a cost ratio of 10 to 1 between the sources and proceed as follows: we exclude the composition with the largest bulk modulus from the HF dataset, take 20 and 10 samples from, respectively, the HF and LF datasets (SF methods only use HF samples), and then initiate the BO methods. We repeat this process 15 times for each BO method to quantify its robustness to the random initial data.

Our second problem concerns hybrid organic–inorganic perovskite (`HOIP`) crystals where the goal is to find the compound with the smallest inter-molecular binding energy. There are one HF and two LF datasets in this problem and their sampling costs are 15, 10, and 5, respectively.[14] The three datasets have similar dimensionalities (1 output

[14] We assign these sampling costs randomly to the LF sources as we do not know which one is more accurate.



(a) `Double-well Potential` **(b)** `Rosenbrock`

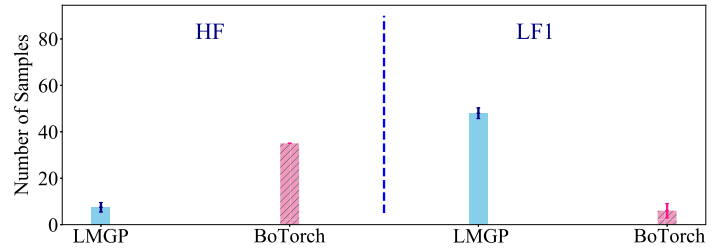
(c) `Borehole` **(d)** `Wing`

Fig. 9. Number of iterations at convergence: As expected, `BoTorch` and $\texttt{LMGP}_{\texttt{CA}}$ need more iterations to converge compared to the SF methods. However, the difference in the case of $\texttt{LMGP}_{\texttt{CA}}$ is very small as our method is quite efficient in leveraging the LF sources. It is noted that, since only one sample is obtained per iteration, these plots are also representative of the total number of samples collected via BO.

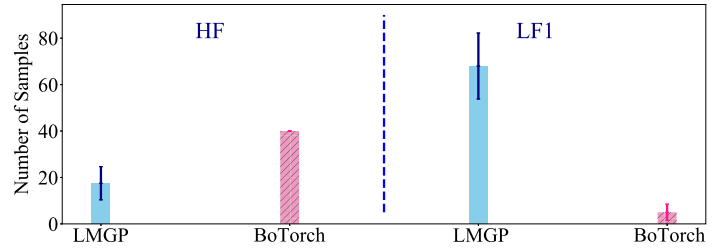
and 3 categorical inputs with 10, 3, and 16 levels) but are of different sizes: the HF dataset has 480 samples, while the first and second LF sources have 179 and 240 samples, respectively. We apply the three BO methods to this problem as follows: we first exclude the best compound from the HF dataset and build the initial MF data by randomly sampling from the three datasets. Then, we launch the BO process where $\texttt{LMGP}_{\texttt{EI}}$ and $\texttt{LMGP}_{\texttt{PI}}$ only use the HF samples. We set the size of the initial datasets to (15, 20, 15) for the HF and LF sources, respectively, and repeat the BO process 15 times to assess the repetition-size variability.

Per Step 0 in Algorithm 2, we train an LMGP to the initial data in each problem to determine whether any of the LF sources must be excluded from BO. As demonstrated in Fig. 12(a), the LF and HF sources in `NTA` are highly correlated since their corresponding latent points are very close in the learnt fidelity manifold of LMGP. However, the latent points in Fig. 12(b) are quite distant and hence we exclude both LF sources from the BO process. It is noted that (1) we provide these manifolds for a randomly selected repetition in each example since they insignificantly change across the repetitions (most changes are due to rotation and translation of all the latent points which do not affect the relative distances), and (2) even though small initial data is used in training the LMGPs, the resulting manifolds provide trustworthy representations of the relative fidelities. To test this second point, we fit an LMGP to the entire data in each example and visualize the resulting manifolds, see Figs. 12(c) and 12(d). As it can be observed, while the manifolds do not match exactly, the relative distances between the latent points are similar.

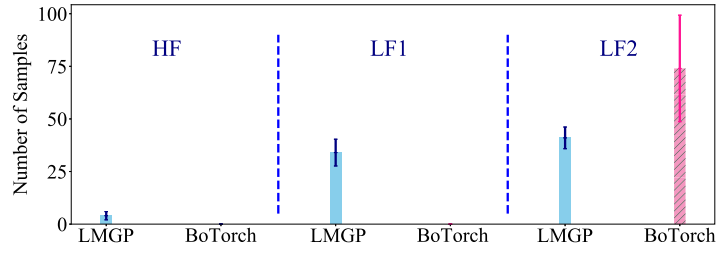
Fig. 13 summarizes the convergence histories by tracking the best HF estimate found by each method (i.e., y_l^* in Eq. (30)) as a function of the accumulated costs. As expected, $\texttt{LMGP}_{\texttt{CA}}$ outperforms the two SF methods in `NTA` but not in `HOIP`. In Fig. 13(a) we observe that $\texttt{LMGP}_{\texttt{EI}}$ and $\texttt{LMGP}_{\texttt{PI}}$ cannot find the optimum compound before the convergence criteria terminate the optimization process. However, these two methods perform quite well in



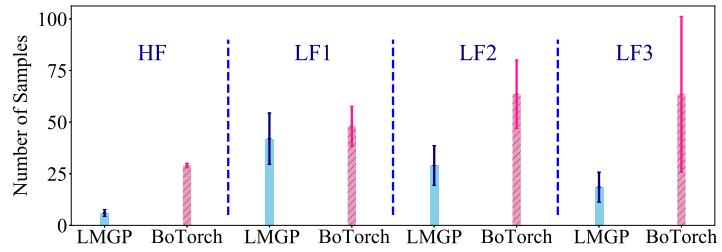
(a) `Double-well Potential`



(b) `Rosenbrock`



(c) `Borehole`

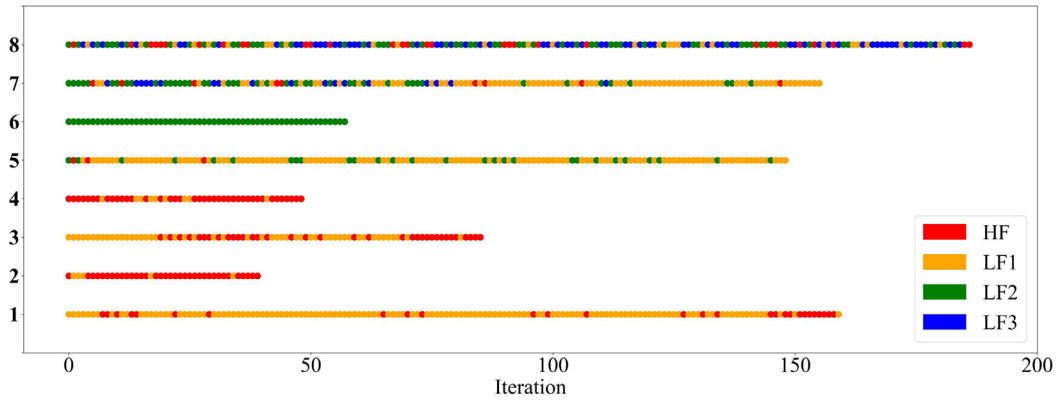


(d) `Wing`

Fig. 10. Number of samples taken from each source: As opposed to `BoTorch`, $\texttt{LMGP}_{\texttt{CA}}$ optimally and automatically adjust the sampling frequency from each source. For instance, `BoTorch` does not sample from two sources in `Borehole` since they are much more expensive than the second LF source. However, $\texttt{LMGP}_{\texttt{CA}}$ not only samples from all sources, but also adjusts the sampling frequency from the LF sources based on their relative accuracy, initial data, and cost (note that LF2 is ten times cheaper to query than LF1 in `Borehole`).

`HOIP` and converge to a value that is very close to the ground truth (the small difference can be eliminated by relaxing the convergence metrics). As expected, $\texttt{LMGP}_{\texttt{CA}}$ finds a sub-optimal compound in `HOIP` since the highly biased LF sources steer the search process in the wrong direction.

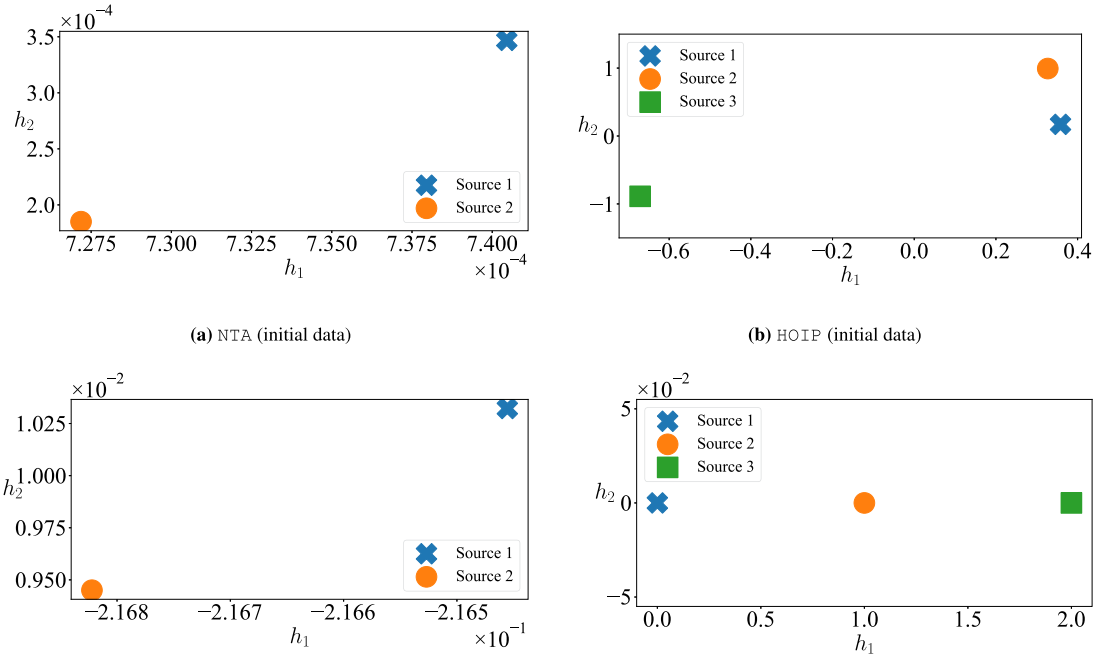
Similar to Section 4.1 we also provide the accumulated cost up to and including the iteration at which each method finds its best compound (which may not correspond to the ground truth) in each example, see Fig. 14. In



1: Double-well Potential (LMGP_{CA}) **3**: Rosenbrock (LMGP_{CA}) **5**: Borehole (LMGP_{CA}) **7**: Wing (LMGP_{CA})

2: Double-well Potential (BoTorch) **4**: Rosenbrock (BoTorch) **6**: Borehole (BoTorch) **8**: Wing (BoTorch)

Fig. 11. Source-wise sampling orders: A repetition is randomly selected from each example to visualize the sampling orders made by $\texttt{LMGP}_{\texttt{CA}}$ and `BoTorch`. The horizontal axis enumerates the number of BO iterations while the vertical axis denotes the example and the MF BO method used for optimization. There are 2, 2, 3 and 4 data sources in each example from top to bottom. Unlike `BoTorch`, $\texttt{LMGP}_{\texttt{CA}}$ balances exploration and exploitation throughout the optimization process.



(a) `NTA` (initial data)

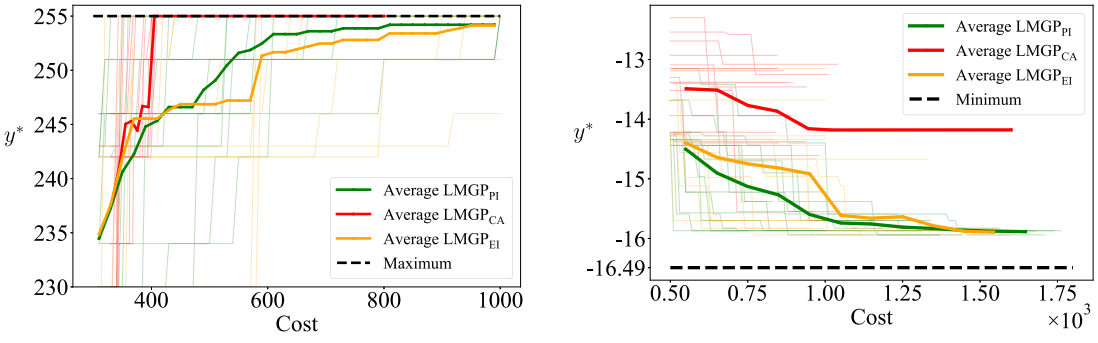
(b) `HOIP` (initial data)

(c) `NTA` (all data)

(d) `HOIP` (all data)

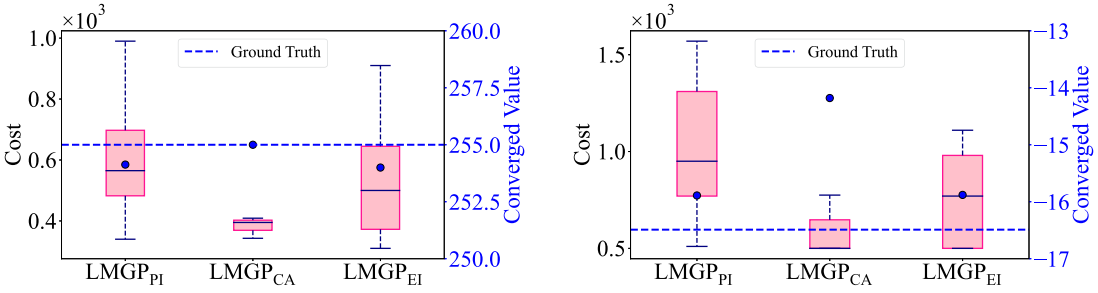
Fig. 12. Fidelity manifolds: The manifolds in the top row are learned via the initial data and indicate that the LF sources should not be used in BO for `HOIP` because the latent points corresponding to them are positioned far from the point encoding the HF source. The manifolds in the second row are built using the entire MF data in each example. The similarity between the two fidelity manifolds of each example indicates that LMGP can effectively learn source-wise discrepancies via small data.

the case of `NTA`, $\texttt{LMGP}_{\texttt{CA}}$ outperforms both $\texttt{LMGP}_{\texttt{EI}}$ and $\texttt{LMGP}_{\texttt{PI}}$ in terms of both accuracy (i.e., finding the ground truth — compare the blue dots to the horizontal dashed line) and consistency (i.e., showing small variations across



(a) NTA **(b)** HOIP

Fig. 13. Convergence history: The plots illustrate the best HF sample (i.e., y_l^* in Eq. (30)) found by each method as a function of sampling costs accumulated during the BO iterations (the cost of initial data is included). As expected, LMGP$_{CA}$ outperforms the single-fidelity methods only in NTA since the LF sources of HOIP have major discrepancies. The solid thick curves indicate the average behavior across the 20 repetitions.



(a) NTA **(b)** HOIP

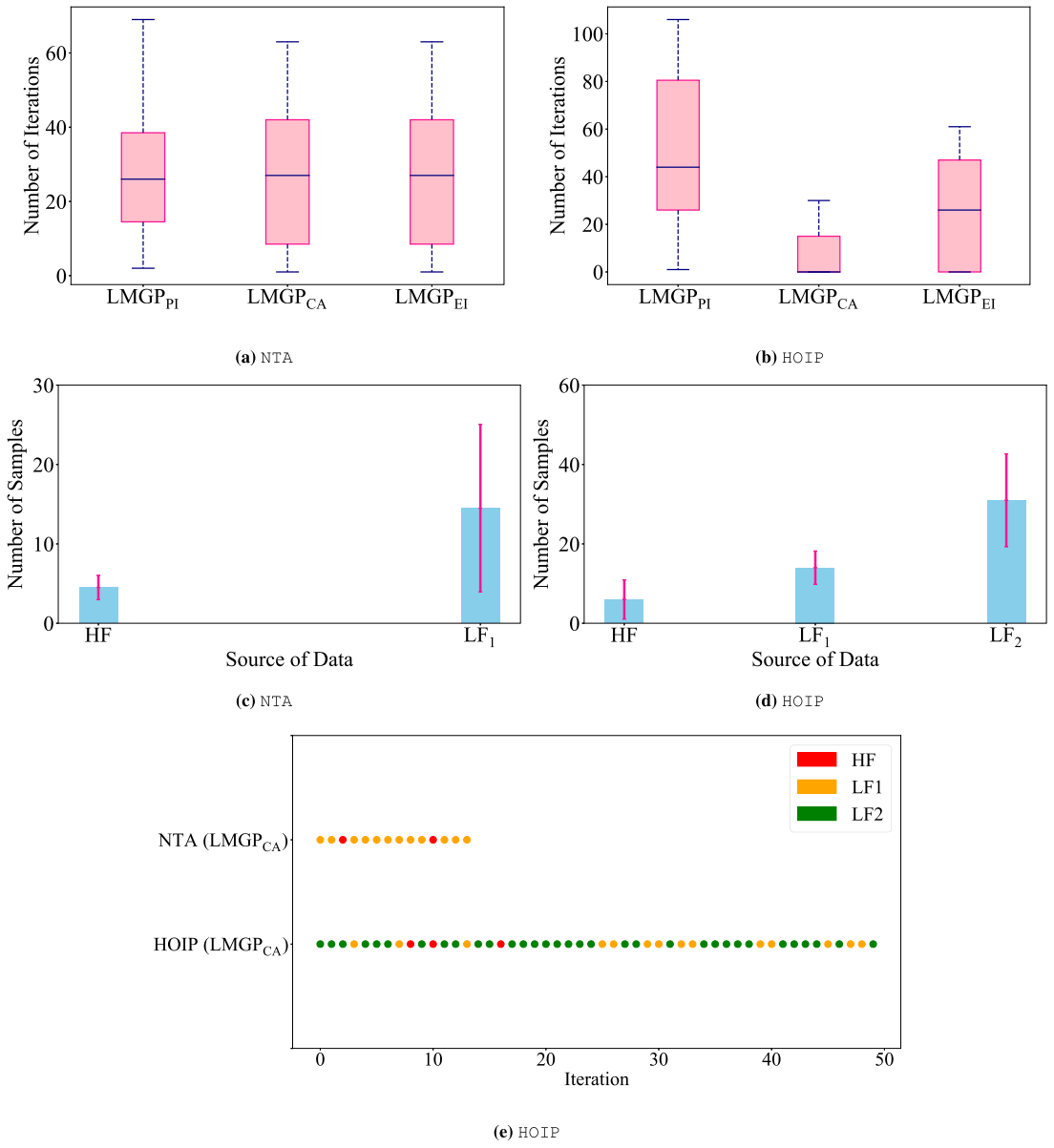
Fig. 14. Accumulated costs before improvements plateau: The box-plots illustrate the accumulated costs up to and including the iteration at which the best HF sample is first obtained (i.e., these box-plots do not consider termination criteria). On the right axis, the converged solution (averaged across the 20 repetitions) and ground truth are demonstrated via, respectively, the blue marker and the horizontal dashed line.

the repetitions — compare the box heights). In the case of HOIP, however, LMGP$_{CA}$ provides lower accuracy than the SF methods since it is using highly biased LF sources. Even though LMGP$_{CA}$ is more robust to variations in the initial data, the lower accuracy does not justify its use for HOIP.

We now investigate the resource allocation behavior of LMGP$_{CA}$. As shown in Figs. 15(a) and 15(b), LMGP$_{CA}$ takes equal or fewer iterations to converge (note that since one sample is taken per iteration, this means that LMGP$_{CA}$ takes fewer overall samples). In the case of NTA, this behavior is desirable especially since most samples are taken from the LF source which is cheaper to query, see Fig. 15(c). However, in the case of HOIP, this seemingly desirable behavior results in convergence to an incorrect solution. Hence, we emphasize the importance of Step 0 in Algorithm 2: while LMGP$_{CA}$ can effectively allocate resources based on the initial dataset sizes and data collection costs (see Figs. 15(d) and 15(e)), highly biased LF sources can steer the search in the wrong direction and, in turn, result in convergence to an incorrect solution.

5. Conclusion

We introduce a multi-fidelity cost-aware framework for Bayesian optimization of expensive black-box functions. Compared to single-source BO, our framework provides improved accuracy and convergence rate by leveraging inexpensive LF sources during the optimization. Unlike existing MF BO techniques, our method accommodates an arbitrary number of LF sources and can effectively balance exploration and exploitation regarding both the search



(a) NTA **(b)** HOIP **(c)** NTA **(d)** HOIP **(e)** HOIP

Fig. 15. Sampling behavior of single- and multi-fidelity BO: In these examples $\texttt{LMGP}_{\texttt{CA}}$ takes no more iterations (i.e., total number of additional samples of any fidelity) than either $\texttt{LMGP}_{\texttt{PI}}$ or $\texttt{LMGP}_{\texttt{EI}}$. In both examples, most of these samples are from LF datasets which is desirable in MF BO as long as the LF sources are sufficiently correlated with the HF source.

space and source utilization. We demonstrate these benefits on both analytic and engineering examples and argue that they are the results of our new acquisition function as well as integrating LMGPs with BO.

One of the major outcomes of our work is determining (only via the initial data) if using LF sources in BO improves the performance. Currently, we make this decision by inspecting the learnt fidelity manifold of LMGP: if the point representing an LF source is far from the point which encodes the HF source, then that LF source should not be used in MF BO. This distance is directly related to the global correlation between an LF and the HF sources and we use this relation to judge whether the discrepancy is large enough. While this simple approach works quite well, it may provide sub-optimal results and hence we plan to improve it in two major directions. Firstly, we envision developing a local metric which enables LF sources to contribute to BO even if they are only correlated

with the HF source on a small portion of the search space. Secondly, we plan to integrate the fidelity metrics with the AFs to scale the information values based on the sample fidelity. With these additions, all LF sources are kept in the loop since they may provide locally useful predictions.

Our new AF does not have any calibration parameters but one can certainty scale its individual components to prioritize (based on, e.g., prior knowledge) sampling from specific sources. There is also potential in designing new utility functions that, in addition to (in lieu of) expected and probability of improvement, are inspired by other AFs such as upper confidence bound. The examples of this papers do not explore these options since we observe high performance (which is much better than the competing methods). In addition, in our studies we use a very simple mechanism for encoding the fidelity via LMGP and assume the data collection costs are given and fixed but these choices and parameters can be adjusted based on the application. We plan to study these directions in our future works.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Code and data availability

The datasets corresponding to analytic and engineering examples are available on the GitLab repository. The source codes for reproducing the results of `BoTorch` are also available at https://gitlab.com/S3anaz/multi-fidelity-cost-aware-bayesian-optimization/-/tree/main/Notebooks.

Acknowledgments

We appreciate the support from National Science Foundation (award numbers OAC-2211908 and OAC-2103708) and the Early Career Faculty grant from NASA's Space Technology Research Grants Program (award number 80NSSC21K1809).

Appendix A. Formulation of EI and PI

We first derive the AF for PI and then follow a similar procedure for EI. We insert PI's utility function in Eq. (11):

$$\alpha_{\mathrm{PI}}(\boldsymbol{x}) = \mathbb{E}[I(\boldsymbol{x}) \mid \mathcal{D}] = \int_{-\infty}^{\infty} \Pr(I(\boldsymbol{x}))I(\boldsymbol{x})\mathrm{d}x \tag{A.1}$$

Eq. (12) demonstrates that $I(\boldsymbol{x})$ is zero for $y(\boldsymbol{x}) < y^*$, so:

$$\alpha_{\mathrm{PI}}(\boldsymbol{x}) = \mathbb{E}[I(\boldsymbol{x}) \mid \mathcal{D}] = \int_{y^*}^{\infty} \Pr(I(\boldsymbol{x}) > 0)\mathrm{d}y \tag{A.2}$$

Hence, to calculate α_{PI} we only need to find $\Pr(I(\boldsymbol{x}) > 0)$ which is a function of the random variable $y(\boldsymbol{x})$. In a GP, the response $y(\boldsymbol{x})$ follows a normal distribution with mean $\mu(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$:

$$y(\boldsymbol{x}) \sim \mathcal{N}\left(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})\right) \tag{A.3}$$

We now apply the reparameterization trick to $y(\boldsymbol{x})$ to calculate PI. Considering $z \sim \mathcal{N}(0, 1)$, then $y(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})z$ is a normal distribution with mean $\mu(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$. Then:

$$\Pr(I(\boldsymbol{x}) > 0) \Leftrightarrow \Pr(y^* < y(\boldsymbol{x})) = \Pr(\frac{y^* - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})} < \frac{y(\boldsymbol{x}) - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}) \tag{A.4}$$

defining $z_0 = \frac{y^* - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}$ which follows $\mathcal{N}\left(0, \sigma^2\right)$ simplifies the above as:

$$\Pr(\frac{y^* - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})} < z) = 1 - \Pr(z \leq \frac{y^* - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})}) = 1 - \Phi(z) = \Phi(-z) = \Phi(\frac{\mu(\boldsymbol{x}) - y^*}{\sigma(\boldsymbol{x})}) \tag{A.5}$$

where Φ is the CDF of the standard normal variable [41,42].

In the case of EI, we follow similarly and use the reparameterization trick to define $y(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})z$ to rewrite Eq. (14) as $I(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})z - y^*$ where z is the standard normal random variable. We now insert this utility function into Eq. (11):

$$\alpha_{EI}(\boldsymbol{x}) \equiv \mathbb{E}[I(\boldsymbol{x})|D] = \int_{-\infty}^{\infty} \max(\mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})z - y^*, 0)\phi(z)\mathrm{d}z \tag{A.6}$$

To eliminate the *max* operator and simplify the integration, we divide $\mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})z - y^*$ into two negative and positive components by finding the switch point:

$$y(\boldsymbol{x}) = y^* \Rightarrow \mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})z = y^* \Rightarrow z_0 = \frac{y^* - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})} \tag{A.7}$$

choosing z_0 as our switch point converts the integration to:

$$\alpha_{EI}(\boldsymbol{x}) = \underbrace{\int_{-\infty}^{z_0} (\mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})z - y^*)\phi(z)\mathrm{d}z}_{\text{zero since } z<z_0, I(\boldsymbol{x})=0} + \int_{z_0}^{\infty} (\mu(\boldsymbol{x}) + \sigma(\boldsymbol{x})z - y^*)\phi(z)\mathrm{d}z \tag{A.8}$$

substituting $\phi(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2)$ in Eq. (A.8) the integrating provides the AF:

$$\begin{aligned}
\alpha_{EI}(\boldsymbol{x}) &= \int_{z_0}^{\infty} \left(\mu(\boldsymbol{x}) - y^*\right) \phi(z)\mathrm{d}z + \int_{z_0}^{\infty} \sigma(\boldsymbol{x}) z \frac{1}{\sqrt{2\pi}} e^{-z^2/2}\, \mathrm{d}z \\
&= \left(\mu(\boldsymbol{x}) - y^*\right) \underbrace{\int_{z_0}^{\infty} \phi(z)\mathrm{d}z}_{1-\Phi(z_0)\equiv 1-\mathrm{CDF}(z_0)} + \frac{\sigma(\boldsymbol{x})}{\sqrt{2\pi}} \int_{z_0}^{\infty} z e^{-z^2/2}\, \mathrm{d}z \\
&= \left(\mu(\boldsymbol{x}) - y^*\right) (1 - \Phi(z_0)) - \frac{\sigma(\boldsymbol{x})}{\sqrt{2\pi}} \int_{z_0}^{\infty} \left(e^{-z^2/2}\right)' \mathrm{d}z \\
&= \left(\mu(\boldsymbol{x}) - y^*\right) (1 - \Phi(z_0)) - \frac{\sigma(\boldsymbol{x})}{\sqrt{2\pi}} \left[e^{-z^2/2}\right]_{z_0}^{\infty} \\
&= \left(\mu(\boldsymbol{x}) - y^*\right) \underbrace{(1 - \Phi(z_0))}_{\Phi(-z_0)} + \sigma(\boldsymbol{x})\phi(z_0)
\end{aligned} \tag{A.9}$$

or:

$$\alpha_{EI}(\boldsymbol{x}) = (\mu(\boldsymbol{x}) - y^*)\Phi(\frac{\mu(\boldsymbol{x}) - y^*}{\sigma(\boldsymbol{x})}) + \sigma(\boldsymbol{x})\phi(\frac{\mu(\boldsymbol{x}) - y^*}{\sigma(\boldsymbol{x})}) \tag{A.10}$$

Appendix B. Fidelity kernels of single-task multi-fidelity GP

Single-task multi-fidelity GP (STGP) uses two fidelity features (1) the data fidelity parameter, x_a, which distinguishes between different fidelity sources, and (2) iteration fidelity parameter, x_b, which is optional and usually exists in hyperparameter tuning problem. These two features are used in $e_i(\cdot)$ which are user-defined functions that model the cross-source correlations in Eq. (19). The formulation of these functions is as follows:

$$e_1(x_a, x_a') = (1 - x_a)(1 - x_a')(1 + x_a x_a')^p \tag{B.1}$$

where p is the degree of polynomial (which needs to be estimated) and has a Gamma prior. e_3 is defined similarly but for the second fidelity:

$$e_3(x_b, x_b') = (1 - x_b)(1 - x_b')(1 + x_b x_b')^p \tag{B.2}$$

Finally, e_2 is the interaction term with four deterministic terms and one polynomial kernel:

$$e_2([x_a, x_b]^T, [x_a', x_b']^T) = (1 - x_b)(1 - x_b')(1 - x_a)(1 - x_a')(1 + [x_a, x_b]^T [x_a', x_b']^T)^p \tag{B.3}$$

Appendix C. Comparison of MF emulators

The large variations in MSEs of LMGP shown in Fig. 3 are due to the *log* scale representation which magnifies small values. Fig. 16 illustrates that in the original space, LMGP has the least MSEs and also variations.

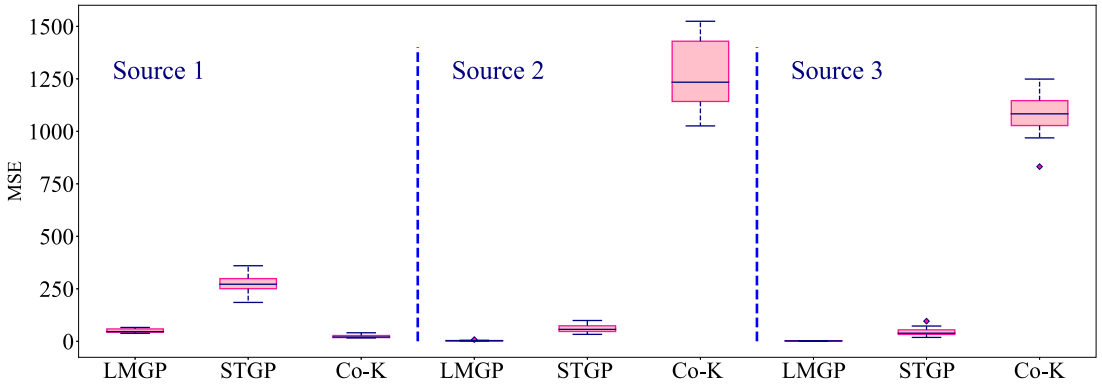


Fig. 16. Emulator comparison in $non-log$ scale: LMGP outperforms other methods as it has the least MSEs and variation in prediction accuracy of all the sources.

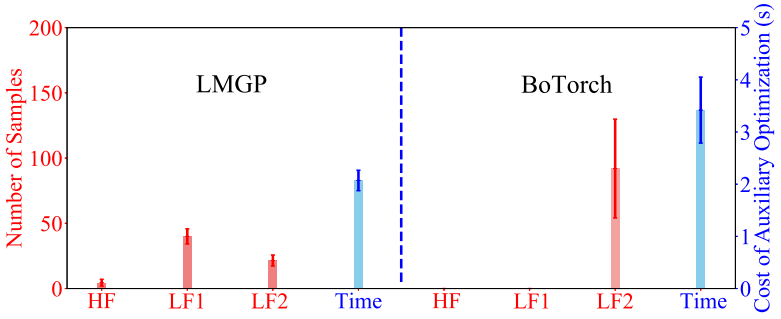


Fig. 17. Computational cost of optimization in $LMGP_{CA}$ and $BoTorch$: Blue bars represent time of the optimization and other bars show the number of samples taken from each source. Despite the larger number of auxiliary optimizations done by $LMGP_{CA}$, its cost is lower than that of $BoTorch$ and its performance is better. Note that the computational cost refers to the auxiliary optimizations and is not sampling cost.

Appendix D. Computational cost of auxiliary optimization

The computational costs of BO largely depend on solving the auxiliary optimization problems. Assuming there are ds data sources, ds independent auxiliary optimizations are solved in $LMGP_{CA}$ while in $BoTorch$ all the sources are optimized jointly. Fig. 17 demonstrates the optimization time of `Borehole` example through 10 repetitions with $LMGP_{CA}$ and $BoTorch$. As shown in this figure, while more auxiliary optimizations are solved in $LMGP_{CA}$, its computational cost is lower than $BoTorch$. This trend is due to the fact that $BoTorch$ uses KG (which is look-ahead and expensive to evaluate) while our proposed AFs are based on EI and PI (which do not require Monte Carlo approximations).

The motivation for using look-ahead AFs despite their high computational cost is to add more flexibility to the model to be able to sample more efficient points. Regarding Fig. 17, our proposed myopic AFs mostly sample from less-expensive LF sources to reduce the uncertainty of the domain, then find the optimum through a few HF samples while KG is misled by the large cost difference among data sources and only samples from the less expensive and less accurate data source. Therefore, $LMGP_{CA}$ outperforms $BoTorch$ in terms of convergence value and computational cost.

Appendix E. Table of numerical examples

Table 1 lists the analytic functions studied in Section 4.1. The error of each LF source with respect to the corresponding HF source is calculated via relative root mean squared error (RRMSE):

$$RRMSE = \sqrt{\frac{(\boldsymbol{y}_l - \boldsymbol{y}_h)^T(\boldsymbol{y}_l - \boldsymbol{y}_h)}{10\,000 \times var(\boldsymbol{y}_h)}} \tag{E.1}$$

Table 1
List of analytic functions: The examples have a diverse degree of dimensionality, number of sources, and complexity. n denotes the number of initial samples and the relative root mean squared error (RRMSE) of an LF source is calculated by comparing its output to that of the HF source at 10 000 random points, see Eq. (E.1). For `Borehole`, LF3 and LF4 become the first and second LF sources, respectively, once LMGP identifies that the listed LF1 and LF2 in this table are highly biased.

Name	Source ID	Formulation	n	RRMSE	Cost
`Double-well Potential`	HF	$0.6x^4 - 0.3x^3 - 3x^2 + 2x$	5	–	1000
	LF	$0.6x^4 - 0.3x^3 - 3x^2 - 1.2x$	0	1.14	1
`Rosenbrock`	HF	$(1-x_1)^2 + 100(x_2 - x_1^2)^2 - 456.3$	5	–	1000
	LF	$(1-x_1)^2 + 100$	10	1.42	1
`Borehole`	HF	$\frac{2\pi T_u(H_u - H_l)}{\ln(\frac{r}{r_w})(1+\frac{2LT_u}{\ln(\frac{r}{r_w})r_w^2 k_w}+\frac{T_u}{T_l})}$	5	–	1000
	LF1	$\frac{2\pi T_u(H_u - 0.8H_l)}{\ln(\frac{r}{r_w})(1+\frac{1LT_u}{\ln(\frac{r}{r_w})r_w^2 k_w}+\frac{T_u}{T_l})}$	5	4.40	100
	LF2	$\frac{2\pi T_u(H_u - H_l)}{\ln(\frac{r}{r_w})(1+\frac{8LT_u}{\ln(\frac{r}{r_w})r_w^2 k_w}+0.75\frac{T_u}{T_l})}$	50	1.54	10
	LF3	$\frac{2\pi T_u(1.09H_u - H_l)}{\ln(\frac{4r}{r_w})(1+\frac{3LT_u}{\ln(\frac{r}{r_w})r_w^2 k_w}+\frac{T_u}{T_l})}$	5	1.30	100
	LF4	$\frac{2\pi T_u(1.05H_u - H_l)}{\ln(\frac{2r}{r_w})(1+\frac{3LT_u}{\ln(\frac{r}{r_w})r_w^2 k_w}+\frac{T_u}{T_l})}$	50	1.3	10
`Wing`	HF	$0.36 s_w^{0.758} w_{fw}^{0.0035} (\frac{A}{\cos^2(\Lambda)})^{0.6} q^{0.006} \times \lambda^{0.04} (\frac{100 t_c}{\cos(\Lambda)})^{-0.3} (N_z W_{dg})^{0.49} + s_w w_p$	5	–	1000
	LF1	$0.36 s_w^{0.758} w_{fw}^{0.0035} (\frac{A}{\cos^2(\Lambda)})^{0.6} q^{0.006} \times \lambda^{0.04} (\frac{100 t_c}{\cos(\Lambda)})^{-0.3} (N_z W_{dg})^{0.49} + w_p$	5	0.19	100
	LF2	$0.36 s_w^{0.8} w_{fw}^{0.0035} (\frac{A}{\cos^2(\Lambda)})^{0.6} q^{0.006} \times \lambda^{0.04} (\frac{100 t_c}{\cos(\Lambda)})^{-0.3} (N_z W_{dg})^{0.49} + w_p$	10	1.14	10
	LF3	$0.36 s_w^{0.9} w_{fw}^{0.0035} (\frac{A}{\cos^2(\Lambda)})^{0.6} q^{0.006} \times \lambda^{0.04} (\frac{100 t_c}{\cos(\Lambda)})^{-0.3} (N_z W_{dg})^{0.49}$	50	5.75	1

where $\boldsymbol{y}_l$ and $\boldsymbol{y}_h$ are vectors of size 10000×1 that store random samples taken from the LF and HF sources, respectively.

Appendix F. Effect of dataset sizes

In practice, the number of initial samples may impact the efficiency of MF BO. In this paper, we initialize BO with dataset sizes that are small given the dimensionality of the problem. To assess the sensitivity of our BO framework to the size of the initial data, we re-evaluate the `Borehole` example of Section 4 with four different initialization (we exclude the two highly biased LF sources). The details about the different initializations are presented in Table 2.

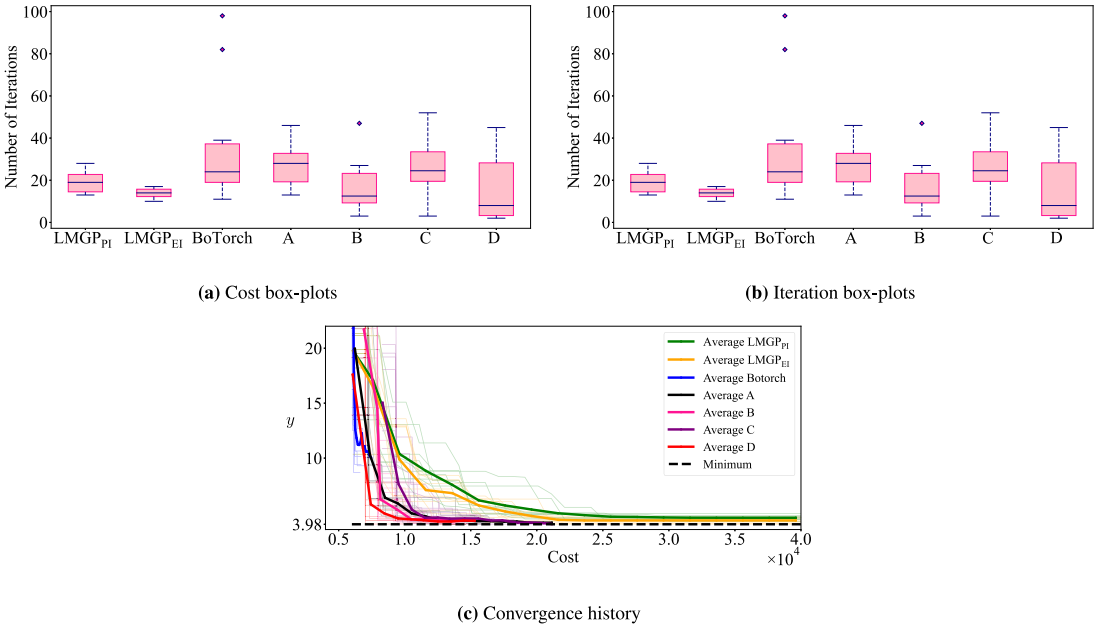
The results are summarized in Fig. 18 and demonstrate that across the four cases $LMGP_{CA}$ has almost the same performance and converges to the ground truth with the least cost compared to the other methods. These results illustrate that the effects of the initial data on the performance of $LMGP_{CA}$ are negligible.

Appendix G. Effect of highly biased low-fidelity sources

In Section 4.1, we exclude two LF sources from `Borehole` due to their high discrepancy with respect to the HF source, see Fig. 6. Below, we summarize the performance of $\texttt{LMGP}_{\texttt{CA}}$ on this problem without removing these two sources. As it can be observed in Fig. 19(a), the optimization is terminated based on the second convergence metric which caps the maximum number of iterations without improvement in the best HF sample (i.e., y_l^* in Eq. (30)).

Table 2
Borehole example with different initialization: To assess the performance of the proposed MFCA BO approach under different initializations, the `Borehole` example is re-evaluated with four different initial data. Then, each example is run 10 times to guarantee the stability of the results. The column numbers indicate the number of initial samples from any source in each scenario.

Example	Initial data		
	HF	LF1	LF2
A	5	10	20
B	5	15	40
C	7	10	30
D	5	5	50

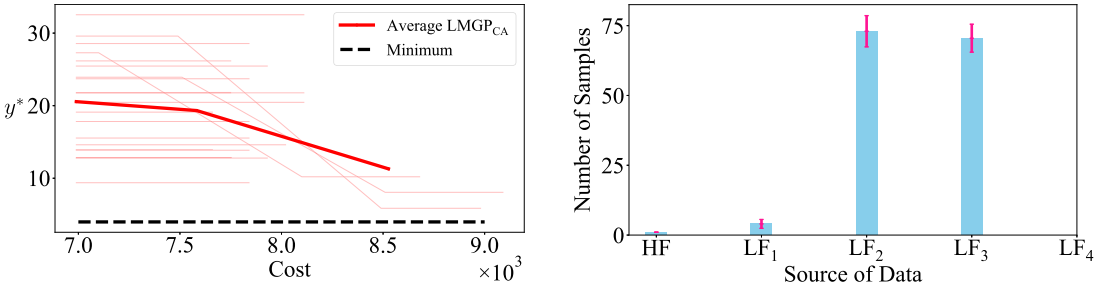


(a) Cost box-plots

(b) Iteration box-plots

(c) Convergence history

Fig. 18. Borehole with different initialization: To assess the sensitivity of the proposed method to the initial data, the `Borehole` example is re-evaluated with four different initialization (A, B, C, D). In all different initializations, $LMGP_{CA}$ converged to the ground truth with the minimum cost which illustrates the negligible sensitivity of $LMGP_{CA}$ to the number of initial data.



(a) Convergence history

(b) Frequency of source utilization

Fig. 19. Effect of highly biased and inexpensive low-fidelity sources: While `LMGP`$_{CA}$ effectively samples from all sources (considering their costs and contribution to the initial data), it converges to an incorrect solution (11.297 while the ground truth is 3.98) since LF1 and LF2 are highly biased. The initial data are not included in 19(b).

Appendix H. List of abbreviations and symbols

Abbreviation	Explanation
AF	Acquisition Function
BO	Bayesian optimization
`BoTorch`	Multi-fidelity BO with BoTorch
Co-K	Co-Kriging
EI	Expected Improvement
GP	Gaussian Process
HF	High-fidelity
HOIP	Hybrid Organic–Inorganic Perovskite
KG	Knowledge Gradient
LF	Low-Fidelity
LMGP	Latent Map Gaussian Process
$\texttt{LMGP}_{\texttt{CA}}$	Proposed MFCA BO approach
$\texttt{LMGP}_{\texttt{EI}}$	Single-fidelity BO whose emulator and AF are LMGP and EI, respectively
$\texttt{LMGP}_{\texttt{PI}}$	Single-fidelity BO whose emulator and AF are LMGP and PI, respectively
MF	Multi-Fidelity
MFCA	Multi-Fidelity Cost-Aware
MLE	Maximum Likelihood Estimation
MLEI	Most Likely Expected Improvement
MSE	Mean Squared Error
NTA	Nanolaminate Ternary Alloy
PI	Probability of Improvement
RRMSE	Relative Root Mean Squared Error
SF	Single-Fidelity
STGP	Single-Task Multi-Fidelity Gaussian

Symbol	Description
$\boldsymbol{A}$	Rectangular matrix that maps $\boldsymbol{\zeta(t)}$ to $\boldsymbol{z(t)}$
$c(\boldsymbol{x}, \boldsymbol{x}')$	Covariance function
ds	Number of data sources
dt	Dimension of categorical inputs
dz	Dimension of the latent map
dx	Dimension of numerical inputs
$\boldsymbol{h}(s) = [h_1, \ldots, h_{dh}]^T$	Latent representation of data source s
$I(\boldsymbol{x})$	Utility function
$'\boldsymbol{j}'$	Categorical vector of size $n_j \times 1$ whose elements are all set to $'j'$
$\mathcal{N}\left(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})\right)$	Normal distribution with mean $\mu(\boldsymbol{x})$ and standard deviation $\sigma^2(\boldsymbol{x})$
n_j	Number of samples obtained from $s(j)$ (i.e., source j)
l_i	Number of distinct levels in i^{th} categorical input
$\boldsymbol{R}$	Correlation matrix
$r(.,.)$	Parametric correlation function
$s = \{'1', \ldots, 'ds'\}$	Categorical variable whose jth element corresponds to data source j
$\boldsymbol{t}$	Categorical inputs (all except source indicator)
$\boldsymbol{U}_j$	$n_j \times (dx + dt)$ feature matrix obtained from $s(j)$
$\boldsymbol{u}$	Mixed inputs
$\boldsymbol{x}$	Input vector
$y(\boldsymbol{x})$	Output/response
$\boldsymbol{y}_j$	$n_j \times 1$ vector of responses obtained from $s(j)$

Symbol	Description
$\boldsymbol{z(t)}$	Points on latent map corresponding to combination $\boldsymbol{t}$ of the categorical variables
$\alpha(\boldsymbol{x})$	Acquisition Function
$\boldsymbol{\zeta(t)}$	Unique prior vector representation of $\boldsymbol{t}$
$\xi(\boldsymbol{x})$	Zero-mean GP
σ^2	Variance of process
$\Phi(z)$	Cumulative density function (CDF)
$\phi(z)$	Probability density function (PDF)
$\boldsymbol{\Omega}$	diag($\boldsymbol{\omega}$)
$\boldsymbol{\omega}$	Scale parameters
$\otimes$	Kronecker product

References

[1] Ryan-Rhys Griffiths, José Miguel Hernández-Lobato, Constrained Bayesian optimization for automatic chemical design using variational autoencoders, Chem. Sci. 11 (2) (2020) 577–586.

[2] Y. Zhang, D.W. Apley, W. Chen, Bayesian optimization for materials design with mixed quantitative and qualitative variables, Sci. Rep. 10 (1) (2020) 4924, http://dx.doi.org/10.1038/s41598-020-60652-9, ISSN: 2045-2322 (Electronic) 2045-2322 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/32188873.

[3] Yiqun Wang, Akshay Iyer, Wei Chen, James M. Rondinelli, Featureless adaptive optimization accelerates functional electronic materials design, Appl. Phys. Rev. (ISSN: 1931-9401) 7 (4) (2020) 041403.

[4] Anh Tran, Julien Tranchida, Tim Wildey, Aidan P. Thompson, Multi-fidelity machine-learning with uncertainty quantification and Bayesian optimization for materials design: Application to ternary random alloys, J. Chem. Phys. 153 (7) (2020) 074705, http://dx.doi.org/10.1063/5.0015672, URL: https://aip.scitation.org/doi/abs/10.1063/5.0015672.

[5] Elvis Osamudiamhen Ebikade, Yifan Wang, Nicholas Samulewicz, Bjorn Hasa, Dionisios Vlachos, Active learning-driven quantitative synthesis–structure–property relations for improving performance and revealing active sites of nitrogen-doped carbon for the hydrogen evolution reaction, React. Chem. Eng. 5 (12) (2020) 2134–2147.

[6] Jasper Snoek, Hugo Larochelle, Ryan P. Adams, Practical bayesian optimization of machine learning algorithms, Adv. Neural Inf. Process. Syst. 25 (2012).

[7] Benjamin Burger, Phillip M. Maffettone, Vladimir V. Gusev, Catherine M. Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M. Alston, Buyi Li, Rob Clowes, et al., A mobile robotic chemist, Nature 583 (7815) (2020) 237–241.

[8] Takamitsu Matsubara, Yoshihito Funaki, Ming Ding, Tsukasa Ogasawara, Kenji Sugimoto, Data-efficient human training of a care motion controller for human transfer assistant robots using bayesian optimization, in: 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), IEEE, 2016, pp. 606–611.

[9] Llewellyn Morse, Zahra Sharif Khodaei, M.H. Aliabadi, A multi-fidelity modelling approach to the statistical inference of the equivalent initial flaw size distribution for multiple-site damage, Int. J. Fatigue 120 (2019) 329–341.

[10] Kwangkyu Yoo, Omar Bacarreza, M.H. Ferri Aliabadi, Multi-fidelity probabilistic optimisation of composite structures under thermomechanical loading using Gaussian processes, Comput. Struct. 257 (2021) 106655.

[11] Xueguan Song, Liye Lv, Wei Sun, Jie Zhang, A radial basis function-based multi-fidelity surrogate model: exploring correlation between high-fidelity and low-fidelity models, Struct. Multidiscip. Optim. 60 (3) (2019) 965–981.

[12] Jiaqing Kou, Weiwei Zhang, Multi-fidelity modeling framework for nonlinear unsteady aerodynamics of airfoils, Appl. Math. Model. 76 (2019) 832–855.

[13] Carl Edward Rasmussen, Gaussian Processes for Machine Learning, 2006.

[14] Matthew Plumlee, Daniel W. Apley, Lifted Brownian kriging models, Technometrics (ISSN: 0040-1706) 59 (2) (2017) 165–177, http://dx.doi.org/10.1080/00401706.2016.1211555, 1537-2723. URL: <Go to ISI>://WOS:000399588600003.

[15] I. Hassaninia, R. Bostanabad, W. Chen, H. Mohseni, Characterization of the optical properties of turbid media by supervised learning of scattering patterns, Sci. Rep. 7 (1) (2017) 15259, http://dx.doi.org/10.1038/s41598-017-15601-4, ISSN: 2045-2322 (Electronic) 2045-2322 (Linking). URL: https://www.ncbi.nlm.nih.gov/pubmed/29127385.

[16] Siyu Tao, Kohei Shintani, Ramin Bostanabad, Yu-Chin Chan, Guang Yang, Herb Meingast, Wei Chen, Enhanced Gaussian process metamodeling and collaborative optimization for vehicle suspension design optimization, in: ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 2B, American Society of Mechanical Engineers.

[17] Yichi Zhang, Siyu Tao, Wei Chen, Daniel W. Apley, A latent variable approach to Gaussian process modeling with qualitative and quantitative factors, Technometrics (ISSN: 0040-1706) 62 (3) (2019) 291–302, http://dx.doi.org/10.1080/00401706.2019.1638834, 1537-2723. URL: <Go to ISI>://WOS:000481877700001.

[18] Olivier Roustant, Esperan Padonou, Yves Deville, Aloïs Clément, Guillaume Perrin, Jean Giorla, Henry Wynn, Group kernels for Gaussian process metamodels with categorical inputs, SIAM/ASA J. Uncertain. Quantif. (ISSN: 2166-2525) 8 (2) (2020) 775–806.

[19] Qiong Zhang, Peter Chien, Qing Liu, Li Xu, Yili Hong, Mixed-input Gaussian process emulators for computer experiments with a large number of categorical levels, J. Qual. Technol. (ISSN: 0022-4065) (2020) 1–11, http://dx.doi.org/10.1080/00224065.2020.1778431, 2575-6230.

[20] Rohit Tripathy, Ilias Bilionis, Marcial Gonzalez, Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation, J. Comput. Phys. (ISSN: 0021-9991) 321 (2016) 191–223.
[21] Songqing Shan, G. Gary Wang, Metamodeling for high dimensional simulation-based design problems, J. Mech. Des. (ISSN: 1050-0472) 132 (5) (2010).
[22] Liwei Wang, Suraj Yerramilli, Akshay Iyer, Daniel Apley, Ping Zhu, Wei Chen, Scalable Gaussian processes for data-driven design using big data with categorical factors, J. Mech. Des. (ISSN: 1050-0472) 144 (2) (2021) http://dx.doi.org/10.1115/1.4052221.
[23] Conference Paper, 2021, pp. 3133–3141, URL: http://proceedings.mlr.press/v130/stanton21a.html.
[24] Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, Andrew Gordon Wilson, Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration, 2018, arXiv preprint arXiv:1809.11165.
[25] James Hensman, Nicolo Fusi, Neil D. Lawrence, Gaussian processes for big data, 2013, arXiv preprint arXiv:1309.6835.
[26] Ramin Bostanabad, Quantification of Microstructure Induced Uncertainty in Multiscale Materials with Random Processes (Thesis), 2019.
[27] Shan Ba, V. Roshan Joseph, Composite Gaussian process models for emulating expensive functions, Ann. Appl. Stat. (ISSN: 1932-6157) 6 (4) (2012) 1838–1860, http://dx.doi.org/10.1214/12-aoas570, URL: <Go to ISI>://WOS:000314458400021.
[28] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G. Wilson, Eytan Bakshy, BoTorch: A framework for efficient Monte-Carlo Bayesian optimization, Adv. Neural Inf. Process. Syst. 33 (2020).
[29] James T. Wilson, Riccardo Moriconi, Frank Hutter, Marc Peter Deisenroth, The reparameterization trick for acquisition functions, 2017, arXiv preprint arXiv:1712.00424.
[30] Nicholas Oune, Ramin Bostanabad, Latent map Gaussian processes for mixed variable metamodeling, Comput. Methods Appl. Mech. Engrg. 387 (2021) 114128.
[31] N. Oune, R. Bostanabad, Latent map Gaussian processes for mixed variable metamodeling, Comput. Methods Appl. Mech. Engrg. (ISSN: 0045-7825) 387 (2021) 114128, http://dx.doi.org/10.1016/j.cma.2021.114128, URL: <Go to ISI>://WOS:000708647400003.
[32] Jonathan Tammer Eweis-Labolle, Nicholas Oune, Ramin Bostanabad, Data fusion with latent map Gaussian processes, J. Mech. Des. (ISSN: 1050-0472) 144 (9) (2022) 1–41, http://dx.doi.org/10.1115/1.4054520.
[33] https://gpytorch.ai/.
[34] R. Bostanabad, T. Kearney, S.Y. Tao, D.W. Apley, W. Chen, Leveraging the nugget parameter for efficient Gaussian process modeling, Internat. J. Numer. Methods Engrg. (ISSN: 0029-5981) 114 (5) (2018) 501–516, http://dx.doi.org/10.1002/nme.5751, URL: <Go to ISI>://WOS:000428998100002.
[35] Jianjun Wang, Yizhong Ma, Linhan Ouyang, Yiliu Tu, A new Bayesian approach to multi-response surface optimization integrating loss function with posterior probability, European J. Oper. Res. 249 (1) (2016) 231–237.
[36] Raul Astudillo, Peter Frazier, Bayesian optimization of composite functions, in: International Conference on Machine Learning, PMLR, 2019, pp. 354–363.
[37] Michael A. Gelbart, Jasper Snoek, Ryan P. Adams, Bayesian optimization with unknown constraints, 2014, arXiv preprint arXiv:1403.5607.
[38] Jialei Wang, Scott C. Clark, Eric Liu, Peter I. Frazier, Parallel Bayesian global optimization of expensive functions, Oper. Res. 68 (6) (2020) 1850–1865.
[39] Yunxiang Zhang, Xiangyu Zhang, Peter I. Frazier, Two-step lookahead Bayesian optimization with inequality constraints, 2021, arXiv preprint arXiv:2112.02833.
[40] Hao Wang, Bas van Stein, Michael Emmerich, Thomas Back, A new acquisition function for Bayesian optimization based on the moment-generating function, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2017, pp. 507–512.
[41] Donald R. Jones, A taxonomy of global optimization methods based on response surfaces, J. Global Optim. 21 (4) (2001) 345–383.
[42] H.-M. Gutmann, A radial basis function method for global optimization, J. Global Optim. 19 (3) (2001) 201–227.
[43] Matthias Schonlau, William J. Welch, Donald R. Jones, Global versus local search in constrained optimization of computer models, in: Lecture Notes-Monograph Series, JSTOR, 1998, pp. 11–25.
[44] Wenlong Lyu, Pan Xue, Fan Yang, Changhao Yan, Zhiliang Hong, Xuan Zeng, Dian Zhou, An efficient bayesian optimization approach for automated optimization of analog circuits, IEEE Trans. Circuits Syst. I. Regul. Pap. 65 (6) (2017) 1954–1967.
[45] Peter I. Frazier, Warren B. Powell, Savas Dayanik, A knowledge-gradient policy for sequential information collection, SIAM J. Control Optim. 47 (5) (2008) 2410–2439.
[46] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G. Wilson, Eytan Bakshy, BoTorch: a framework for efficient Monte-Carlo Bayesian optimization, Adv. Neural Inf. Process. Syst. 33 (2020) 21524–21538.
[47] Guanghui Zhu, Ruancheng Zhu, Accelerating hyperparameter optimization of deep neural network via progressive multi-fidelity evaluation, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2020, pp. 752–763.
[48] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Joshua Marben, Philipp Müller, Frank Hutter, BOAH: A tool suite for multi-fidelity bayesian optimization & analysis of hyperparameters, 2019, arXiv preprint arXiv:1908.06756.
[49] Hyunghun Cho, Yongjin Kim, Eunjung Lee, Daeyoung Choi, Yongjae Lee, Wonjong Rhee, Basic enhancement strategies when using Bayesian optimization for hyperparameter tuning of deep neural networks, IEEE Access 8 (2020) 52588–52608.
[50] Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, Andrew Gordon Wilson, Practical multi-fidelity Bayesian optimization for hyperparameter tuning, in: Uncertainty in Artificial Intelligence, PMLR, 2020, pp. 788–798.
[51] Tharathep Phiboon, Krittin Khankwa, Nutchanan Petcharat, Nattaphon Phoksombat, Masahiro Kanazaki, Yuki Kishi, Sujin Bureerat, Atthaphon Ariyarit, Experiment and computation multi-fidelity multi-objective airfoil design optimization of fixed-wing UAV, J. Mech. Sci. Technol. 35 (9) (2021) 4065–4072.

[52] Qi Sun, Tinghuan Chen, Siting Liu, Jianli Chen, Hao Yu, Bei Yu, Correlated multi-objective multi-fidelity optimization for hls directives design, ACM Trans. Des. Autom. Electron. Syst. (TODAES) 27 (4) (2022) 1–27.
[53] Syrine Belakaria, Aryan Deshwal, Janardhan Rao Doppa, Multi-fidelity multi-objective Bayesian optimization: An output space entropy search approach, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 10035–10043, 06.
[54] Johan Dahlin, Thomas Bo Schön, Mattias Villani, Approximate Inference in State Space Models with Intractable Likelihoods using Gaussian Process Optimisation, 2014.
[55] Mahdi Imani, Seyede Fatemeh Ghoreishi, Douglas Allaire, Ulisses M. Braga-Neto, MFBO-SSM: Multi-fidelity Bayesian optimization for fast inference in state-space models, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 7858–7865, 01.
[56] Edwin V. Bonilla, Kian Chai, Christopher Williams, Multi-task Gaussian process prediction, Adv. Neural Inf. Process. Syst. 20 (2007).
[57] David A. Stainforth, Tolu Aina, Carl Christensen, Mat Collins, Nick Faull, Dave J. Frame, Jamie A. Kettleborough, S. Knight, A. Martin, J.M. Murphy, et al., Uncertainty in predictions of the climate response to rising levels of greenhouse gases, Nature 433 (7024) (2005) 403–406.
[58] Weizhao Zhang, Ramin Bostanabad, Biao Liang, Xuming Su, Danielle Zeng, Miguel A. Bessa, Yanchao Wang, Wei Chen, Jian Cao, A numerical Bayesian-calibrated characterization method for multiscale prepreg preforming simulations with tension-shear coupling, Compos. Sci. Technol. 170 (2019) 15–24.
[59] Siyu Tao, Daniel W. Apley, Wei Chen, Andrea Garbo, David J. Pate, Brian J. German, Input mapping for model calibration with application to wing aerodynamics, AIAA J. (ISSN: 0001-1452) 57 (7) (2019) 2734–2745, http://dx.doi.org/10.2514/1.J057711, 1533-385X. URL: <Go to ISI>://WOS:000488793600008.
[60] Chengkun Ren, Fenfen Xiong, Fenggang Wang, Bo Mo, Zhangli Hu, A maximum cost-performance sampling strategy for multi-fidelity PC-Kriging, Struct. Multidiscip. Optim. 64 (6) (2021) 3381–3399.
[61] Shishi Chen, Zhen Jiang, Shuxing Yang, Daniel W. Apley, Wei Chen, Nonhierarchical multi-model fusion using spatial random processes, Internat. J. Numer. Methods Engrg. 106 (7) (2016) 503–526.
[62] S. Ashwin Renganathan, Vishwas Rao, Ionel M. Navon, CAMERA: A method for cost-aware, adaptive, multifidelity, efficient reliability analysis, 2022, arXiv preprint arXiv:2203.01436.
[63] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, Nando De Freitas, Taking the human out of the loop: A review of Bayesian optimization, Proc. IEEE 104 (1) (2015) 148–175.
[64] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, Masayuki Karasuyama, Multi-fidelity Bayesian optimization with max-value entropy search and its parallelization, in: International Conference on Machine Learning, PMLR, ISBN: 2640-3498, pp. 9334–9345.
[65] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, Matthias W. Seeger, Information-theoretic regret bounds for gaussian process optimization in the bandit setting, IEEE Trans. Inform. Theory 58 (5) (2012) 3250–3265.
[66] Daniel Russo, Benjamin Van Roy, Learning to optimize via posterior sampling, Math. Oper. Res. 39 (4) (2014) 1221–1243.
[67] José Miguel Hernández-Lobato, Matthew W. Hoffman, Zoubin Ghahramani, Predictive entropy search for efficient global optimization of black-box functions, Adv. Neural Inf. Process. Syst. 27 (2014).
[68] Rémi Pautrat, Konstantinos Chatzilygeroudis, Jean-Baptiste Mouret, Bayesian optimization with automatic prior selection for data-efficient direct policy search, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 7571–7578.
[69] Wolfgang Ponweiser, Tobias Wagner, Dirk Biermann, Markus Vincze, Multiobjective optimization on a limited budget of evaluations using model-assisted S-Metric selection, in: International Conference on Parallel Problem Solving from Nature, Springer, 2008, pp. 784–794.
[70] Yichi Zhang, Daniel W. Apley, Wei Chen, Bayesian optimization for materials design with mixed quantitative and qualitative variables, Sci. Rep. 10 (1) (2020) 1–13.
[71] Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, Si-Hao Deng, Hyperparameter optimization for machine learning models based on Bayesian optimization, J. Electron. Sci. Technol. 17 (1) (2019) 26–40.
[72] Chan-Uk Yeom, Keun-Chang Kwak, Performance evaluation of automobile fuel consumption using a fuzzy-based granular model with coverage and specificity, Symmetry 11 (12) (2019) 1480.