Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.086040

ARTICLE

Multi-Information Source Bayesian Optimization with Location-Varying Costs

Antonio Candelieri[1,*], **Francesco Archetti**[2] **and Iman Seyedi**[2,*]

[1]Department of Economics Management and Statistics, University of Milano-Bicocca, Milan, Italy
[2]Department of Computer Science Systems and Communication, University of Milano-Bicocca, Milan, Italy

*Corresponding Authors: Antonio Candelieri. Email: antonio.candelieri@unimib.it;
Iman Seyedi. Email: seyediman.seyedi@unimib.it

Received: 22 May 2026; Accepted: 25 August 2026; Published: 22 October 2026

ABSTRACT: This paper addresses black-box optimization over multiple information sources whose fidelity and query cost change across the search space. We propose an approach that uses: (i) an Augmented Gaussian Process as a single model of the objective function over the search space and sources, and (ii) a Gaussian Process to model the location-dependent cost of each source. The former is used in a Confidence Bound-based acquisition function to select the next source and location to query, while the latter is used to penalize the value of the acquisition depending on the expected query cost for any source-location pair. Multi-information source optimization (MISO) is characterized by the availability of inexpensive approximations (i.e., sources) of the more expensive objective function. In contrast, cost-aware Bayesian optimization addresses settings where the query cost varies across the search space. When information sources come with explicit information about their approximation quality (aka fidelity), MISO specializes in multi-fidelity optimization. However, most multi-fidelity approaches require hierarchically organized sources and cannot handle location-dependent fidelities. A key contribution of this manuscript is a multi-information source setting in which both fidelity and query cost of the sources are black-box and location-dependent, endowed with a cost and location-aware acquisition function. We provide an empirical evaluation of the proposed method on an AutoML task: hyperparameter optimization of two classifiers on three datasets of different sizes. The acquisition function proposed can be naturally generalized to more complex cost models, including movement costs, which represents an important perspective for multi-source Bayesian optimization.

KEYWORDS: Multi-information source optimization; Bayesian optimization; location-dependent costs; augmented Gaussian process; cost-aware acquisition function

1 Introduction

Bayesian Optimization (BO) [1–3] is a sample-efficient strategy for global optimization of blackbox, expensive, and multi-extremal Functions. The reference global optimization problem is:

$$argmin_{x\in\Omega\subset\mathcal{R}^d} f(x) \tag{1}$$

where Ω is the d-dimensional search space.

BO consists of two key components: a probabilistic surrogate model of $f(x)$, fitted on the function evaluations performed so far, and an acquisition function providing the next location to query, while balancing between exploitation and exploration, depending on the predictions, and associated uncertainty, of the probabilistic surrogate model. Updating the model conditioned on the observations and selecting the

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

next location to query are sequentially iterated until a termination criterion is met, typically a maximum number of function evaluations.

Thanks to its sample efficiency, compared to competing methods, BO is currently the standard technique for Automated Machine Learning (AutoML) [4,5] and also successfully applied in Neural Architecture Search (NAS) [6]. Other relevant applications of BO concern simulation-optimization [7] and control of complex systems [8]. The practical relevance of BO extends further to scientific experimentation, engineering design, and clinical trials, where evaluating a candidate solution may involve running expensive physical simulations, conducting laboratory experiments, or training large machine learning models. These are all scenarios where minimizing the number of function evaluations is critical.

However, the basic BO algorithm does not directly incorporate or utilize any information about query cost, which is crucial in many real-life applications. A compelling example comes from AutoML and NAS: Ref. [9] provides an estimate of the financial and environmental costs associated with optimizing the hyperparameters of deep neural networks in the domain of Natural Language Processing. The astonishing amount of energy required can generate an amount of CO_2 emissions around five times those generated by a car during its own lifetime. This highlights how ignoring query cost in the optimization loop can lead to outcomes that are not only computationally wasteful but also environmentally unsustainable.

Therefore, recent research has been proposing innovative BO-based approaches designed for more challenging settings where information about query cost is explicitly used. This research can be broadly split into two branches: (i) multi-information source optimization (MISO), characterized by the availability of cheap approximations (i.e., sources) of the more expensive $f(x)$, and (ii) cost-aware (Bayesian) optimization, assuming a cost to query $f(x)$, which changes over the search space. While both branches have been studied independently and have demonstrated significant improvements in cost efficiency over standard BO, they have so far been treated as separate research directions.

In practice, however, these two challenges frequently co-occur. Consider, for instance, hyperparameter optimization of a machine learning algorithm across datasets of varying size: cheaper sources correspond to smaller data subsets, but the cost of evaluating a given hyperparameter configuration on any subset is itself location-dependent, as some configurations are computationally far more demanding than others regardless of the dataset size. To the best of our knowledge, no existing approach addresses the setting where both the approximation quality and the query cost of each source vary simultaneously across the search space. This gap motivates the present work.

We propose MISO-wiLDCosts (MISO with Location-Dependent Costs), which jointly models location-dependent fidelity and cost across multiple information sources using an Augmented Gaussian Process (AGP), and selects the next source-location pair to query through a cost- and location-aware acquisition function. The approach is validated on a hyperparameter optimization task involving two ML classifiers and three datasets of different sizes, demonstrating competitive classification accuracy at significantly reduced cumulated query cost.

The rest of the paper is organized as follows. Section 2 reviews the related work, covering multi-information source and multi-fidelity optimization (Section 2.1), cost-aware Bayesian optimization (Section 2.2), and the Gaussian Process regression framework underlying our approach (Section 2.3). Section 3 presents MISO-wiLDCosts, detailing the source and cost modelling (Section 3.1), the Augmented Gaussian Process (Section 3.2), and the acquisition function (Section 3.3). Section 4 describes the experimental setting, and Section 5 reports and discusses the results. Finally, Section 6 draws conclusions, discusses limitations, and outlines future directions.

2 Related Work

This section reviews the main lines of research relevant to the proposed approach, covering multi-information source and multi-fidelity Bayesian optimization in Section 2.1, and cost-aware Bayesian optimization in Section 2.2.

2.1 Multi-Information Source Optimization

As far as MISO is concerned, the problem (1) has to be solved by efficiently using S information sources, $f_1(x), \ldots, f_S(x)$, providing approximations, with different accuracy, of $f(x)$, each one at its own query cost c_s, with $s = 1, \ldots, S$. Thus, accuracy is location-dependent, but cost is not. Efficiently solving (1) means that the query cost cumulated along the optimization process must be kept as low as possible. Sources can always be sorted by decreasing query cost, so that $f_1(x)$ is the most expensive one, usually with $f_1(x) = f(x)$ in the case that $f(x)$ can be directly queried.

When information sources come with explicit information about their approximation quality (aka *fidelity*), MISO specializes in multi-fidelity optimization, first proposed in [10] and more recently addressed in [11–14]. Most multi-fidelity approaches exploit hierarchical relations among sources, based on their fidelities; even if already in [15], some drawbacks were highlighted. More precisely, hierarchical organization requires the assumption that information sources are unbiased, meaning that noise must be independent across sources [16,17]. Moreover, querying a source with a certain fidelity at a given location x implies that no further knowledge can be obtained by querying any other source of lower fidelity, at any location. Consequently, most of the multi-fidelity approaches cannot be applied in the case that sources cannot be hierarchically organized.

In Ref. [16], the first approach for nonhierarchical information sources has been proposed, addressing *location-dependent* fidelities of the sources and defining the more general MISO setting. More recently, Refs. [17,18] have provided improvements to [16]. All these methods are based on the idea of using a separate model for each information source (i.e., a Gaussian Process-GP) and then fusing their predictions and related uncertainties through the method proposed in [19], which became the standard practice for the fusion of normally distributed data. More broadly, the challenge of principled model fusion across heterogeneous data sources is an active research theme, with federated learning representing a parallel paradigm where distributed models are aggregated rather than queried sequentially [20]. In [21], a different procedure has been recently proposed, where instead of fusing GPs, sparsification is used to create the so-called Augmented Gaussian Process (AGP). We used this model—summarized in Section 3.2—and extended MISO-AGP to deal with location-dependent costs.

In the AutoML context, Ref. [22] developed a multi-task Bayesian optimization framework that transfers information across related tasks and demonstrated a cost-sensitive hyperparameter optimization setting in which a small dataset is used to explore hyperparameter configurations for a larger dataset. This work provides an early example of using cheaper related evaluations in AutoML, but it is not intended here as the first general MISO or multifidelity optimization method. More recently, Ref. [23] proposed FABOLAS (Fast Bayesian Optimization on Large dataSets), a hyperparameter optimization (HPO) tool that simultaneously optimizes the hyperparameter values of an ML algorithm and the size of the dataset portion to consider. Also, in [21], an HPO task is considered, with two information sources, related to a large dataset and a small portion of it, respectively. All these approaches resulted in a more cost-efficient approach than using BO for HPO performed on the large dataset only, without significant degradation of the final ML model's accuracy. Alternative budget-aware optimization strategies have also been explored outside the MISO framework, such as hyper-heuristics guided by probabilistic graphical models [24], which reduces evaluation cost through adaptive algorithm selection rather than multi-source approximations. Beyond HPO, MISO-BO has also

been applied to reward design in multi-agent systems [25], where previously learned equilibria serve as biased information sources at the upper level of a bi-level optimization framework, achieving significant computational savings over standard approaches. More recently, multi-source Bayesian optimization has also been applied to source localization problems. Bayesian optimization has also recently been adopted for graph-based multiple-source localization. Zhang et al. [26] formulate localization from a single observation using graph Bayesian optimization, while Zhou et al. [27] extend this direction through graph representation learning for multi-source localization. Zhou et al. [27] combine graph representation learning with Bayesian optimization to efficiently localize multiple sources in complex network environments, further illustrating the versatility of MISO beyond engineering design and AutoML.

A further extension of multi-fidelity BO concerns robustness to unreliable information sources. Mikkola et al. [28] consider a setting with a primary informative source and one or more auxiliary sources whose quality cannot be guaranteed, proposing an approach that makes MFBO robust to the inclusion of such unreliable sources. This is relevant to the broader MISO setting because, in practice, the fidelity of auxiliary sources may be unknown or variable across the search space, motivating methods that treat source reliability as an additional uncertainty to be modelled rather than a fixed assumption. More recently, Jacobs and Ali Javidian [29] proposed incorporating causal principles into multi-source Bayesian optimization, enabling information sharing across sources through causal relationships and further extending the flexibility of the MISO framework.

2.2 Cost-Aware Optimization

Although the quoted MISO and multi-fidelity approaches generalize from sample efficiency to cost efficiency—thus, they are "aware of costs"—they assume that the query cost of any source is constant over the search space. However, this is not true in practice, even using a single source, as empirically demonstrated in [30], according to the distribution of the time required to evaluate a set of 5000 randomly selected hyperparameter configurations, for five common ML algorithms. This is also confirmed in this paper (Section 4.1), relatively to 1000 hyperparameter configurations for two ML algorithms on three datasets.

The seminal work aimed at making BO cost-aware is [31], which proposes to penalize the acquisition function, specifically Expected Improvement (EI), by the location-dependent cost, $c(x)$, leading to the *EI-per-unit of cost*: $EI_{pu}(x) = EI(x)/c(x)$. However, $EI_{pu}(x)$ is basically driven by the query cost, biased towards cheap locations, and, therefore, it performs well only when optima are relatively cheap. To overcome this undesired behaviour, Ref. [30] proposes CArBO (Cost Apportioned BO), consisting of two consecutive stages: (*i*) a cost-effective selection of initial locations to query (i.e., cost-effective initial design) and (*ii*) a cost-cooling strategy where the penalty associated with the cost in $EI_{pu}(x)$ is modulated according to the query cost incurred so far. More precisely, $EI - cool(x) = EI(x)/c(x)^{\alpha}$, with α the cooling factor is defined as $\alpha = (\tau - \tau_n)/(\tau - \tau_{init})$, with τ the overall "budget" (i.e., maximum cumulated query cost), τ_n the cost accumulated up to the current iteration n, and τ_{init} the cost of the initial design. Finally, in CArBO, $c(x)$ is modelled through a warped GP.

Another recent paper [32] has proposed a cost-aware acquisition function based on Information-Directed Sampling (IDS), a principled framework for balancing expected regret and information gain. The original IDS framework was introduced by [33]. The proposed cost-aware acquisition function, namely CostIDS, also balances cost, along with regret and information gain. However, an additional constraint is introduced in optimizing CostIDS to avoid extremely cheap points being chosen repeatedly without any significant increase in information.

A complementary line of work addresses cost variability not at the level of query locations but at the level of which variables are queried. Tay et al. [34] propose a setting where, at each iteration, the learner

selects a subset of input variables and fixes their values, while the remaining variables are randomly sampled. Each chosen subset carries an associated cost, introducing a novel trade-off: querying more variables yields more directed learning but incurs higher cost, whereas leaving variables unobserved reduces cost at the price of less controlled exploration.

The issue of path-dependent costs—where the cost of a query depends not only on the chosen location but also on the trajectory of previous queries—is addressed by Liu et al. [35]. Their approach models the cost-aware acquisition problem as a constrained Markov Decision Process, conducting regret analysis in terms of information gains. They also study the impact of inexact acquisition function optimization, noting that while solving the acquisition problem is generally less demanding than optimizing the objective directly, theoretical guarantees may still require knowledge of the minimizer. This latter issue is further examined by Kim et al. [36], who analyze the effect of inexact acquisition optimization on regret bounds, focusing specifically on GP-UCB and GP-TS (Thompson Sampling). They show that random grid search can serve as a theoretically justified and practically effective acquisition solver, even when the acquisition is not solved to optimality.

A distinct form of location-dependent cost arises when successive queries incur a movement penalty proportional to the distance between consecutive sampling locations. Chen and Al Kontar [37] formalize this as the "traveling bandit" problem, in which the fundamental tension lies between broad exploration—essential for identifying good solutions—and the high movement costs that drastic design changes entail. More recently, Bayesian optimization has also been investigated for resource allocation problems in Wasserstein spaces. Candelieri et al. [38] formulate resource allocation under semi-bandit feedback by exploiting Wasserstein geometry within the Bayesian optimization framework, further illustrating the applicability of cost-aware BO beyond conventional black-box optimization settings. Unlike conventional cost-aware BO, movement cost depends not only on the current design but also on the history of designs chosen by the algorithm, making it inherently path-dependent. This type of cost structure is prevalent in industrial processes and environmental monitoring applications, and represents an important generalization of the standard cost-aware BO setting. Beyond the MISO setting, BO has also been applied to large-scale infrastructure planning under demand uncertainty; Jo et al. [39] propose a BO-based large-neighborhood search framework for multi-period electric vehicle charging station deployment under latent charging demand, illustrating the broader applicability of BO to computationally expensive decision-making problems.

2.3 Gaussian Process Regression

GP modelling [40,41] is a non-parametric kernel-based learning method for probabilistic regression and classification. A GP regression model is a random function $f\colon \Omega \to \mathcal{R}$ with output drawn from a multivariate normal distribution, formally $f(x) \sim N(\mu(x), \sigma^2(x))$. In BO, GP regression is typically used as a probabilistic surrogate model, fitted to the n function evaluations performed so far. Let $X_{1:n} = \{x^1, \ldots, x^n\}$ and $\boldsymbol{y} = \{y^1, \ldots, y^n\}$ denote, respectively, the n locations and the associated observed values, then fitting the GP means to compute the posterior mean and variance of the multivariate normal distribution as follows:

$$\begin{aligned} \mu(x) &= k\left(x, X_{1:n}\right)\left[K + \lambda^2 I\right]^{-1} y \\ \sigma^2(x) &= k\left(x, x\right) - k(x, X_{1:n})\left[K + \lambda^2 I\right]^{-1} k(X_{1:n}, x) \end{aligned} \tag{2}$$

where λ^2 is the variance of a zero-mean Gaussian noise in the case of noisy observations (i.e., $y^i = f(x^i) + \varepsilon$, with $\varepsilon \sim N(0, \lambda^2)$), $K \in \mathcal{R}^{n \times n}$, such that $K_{ij} = k(x_i, x_j)$, with k a kernel function modelling the covariance in the GP. Finally, $k(x, X_{1:n})$ is a vector whose i-th component is given by $k(x, x_i)$ (for completeness, $k(X_{1:n}, x) = k\left(x, X_{1:n}\right)^T$).

The most widely adopted kernels are Squared Exponential (aka Gaussian), Matern, Power Exponential, and Exponential (aka Laplacian). Each kernel implies a different prior on the structural properties of the sample paths of the latent function under the GP, such as differentiability [42]. Moreover, each kernel has its own hyperparameters to adjust the GP's posterior depending on observations: GP's hyperparameters are usually tuned via Maximum Log-likelihood Estimation (MLE) or Maximum A Posteriori estimation (MAP).

In this paper, the Matern 3/2 kernel is adopted:

$$k_{m3/2}(x, \acute{x}) = \sigma_k^2 \left(1 + \frac{\sqrt{3}r}{\ell}\right) e^{-\frac{\sqrt{3}r}{\ell}} \tag{3}$$

with $r = ||x - \acute{x}||$. The kernel's hyperparameters σ_k^2 and ℓ, namely the kernel amplitude and the characteristic length scale, are estimated via MLE.

2.4 Our Contributions

To the best of our knowledge, no prior work has jointly addressed a multi-information source Bayesian optimization setting in which both the approximation quality (fidelity) and the query cost of each source are black-box and location-dependent—that is, neither is known in advance nor assumed constant across the search space.

To address this setting, we propose MISO-wiLDCosts (MISO with Location-Dependent Costs), whose goal is to solve problem (1) while keeping the cumulated query cost as small as possible, given a set of S information sources $\{f_s(x)\}_{s=1:S}$ whose fidelities and query costs $c_{s(x)}$ are both black-box and location-dependent. Specifically, the contributions of this paper are:

An extension of the Augmented Gaussian Process (AGP) model to jointly capture the objective function over the search space and across all information sources whose fidelity and query cost are both black-box and location-dependent, without requiring hierarchical organization of the sources or location-independent cost assumptions.

A cost- and location-aware acquisition function based on Confidence Bounds, which selects the next source-location pair to query by balancing exploration and exploitation while penalizing the acquisition value by the expected query cost, as modelled by a dedicated GP for each source's cost function.

An empirical evaluation on an AutoML task, hyperparameter optimization (HPO) of two ML classifiers on three datasets of different sizes, demonstrating that MISO-wiLDCosts achieves significantly lower cumulated query cost compared to single-source cost-aware BO, at the cost of a small, dataset-dependent degradation in classification accuracy in some settings.

3 Methodology

This section presents the three components of MISO-wiLDCosts, building on the GP framework introduced in Section 2.3, this section presents the three components of MISO-wiLDCosts.

3.1 Modelling Sources and Costs

Three distinct families of Gaussian Process models are used in MISO-wiLDCosts: a separate GP fitted on each information source's function evaluations, the AGP that fuses them into a single model (Section 3.2), and a separate GP fitted on each source's query cost (this section). Let $\boldsymbol{y_s}$ denotes a function evaluation performed on the source s, where $\boldsymbol{y_s} = \boldsymbol{f_s}(\boldsymbol{x}) + \boldsymbol{\varepsilon_s}$ and $\boldsymbol{\varepsilon_s}$ is a zero-mean Gaussian noise associated with that source, formally $\boldsymbol{\varepsilon_s} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{\lambda_s^2})$. Let $\boldsymbol{z_s}$ denotes the (black-box) query cost to observe $\boldsymbol{y_s}$, that is $\boldsymbol{z_s} = \boldsymbol{c_s}(\boldsymbol{x}) + \boldsymbol{\delta_s}$, with $\boldsymbol{\delta} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{\zeta_s^2})$.

It is important to note that, by the definition of query cost, $\boldsymbol{y}_s$ and $\boldsymbol{z}_s$ are not *decoupled*, in the sense that $\boldsymbol{y}_s$ cannot be observed without paying $\boldsymbol{z}_s$, and *vice versa*.

Let $\boldsymbol{D}_s = \left\{\boldsymbol{x}^i, \boldsymbol{y}_s^i, \boldsymbol{z}_s^i\right\}_{i=1,\ldots,n_s}$ denotes the dataset collecting all the relevant information along the optimization process, with n_s the number of function evaluations performed on the source s. The following two projections of $\boldsymbol{D}_s$ are considered:

$$\boldsymbol{F}_s = \left\{\boldsymbol{x}^i, \boldsymbol{y}_s^i\right\}_{i=1,\ldots,n_s} \tag{4}$$

namely the function evaluations dataset, storing locations queried and function values observed, and

$$\boldsymbol{C}_s = \left\{\boldsymbol{x}^i, \boldsymbol{z}_s^i\right\}_{i=1,\ldots,n_s} \tag{5}$$

namely, the query costs dataset, which stores locations queried and query costs.

The two sets, $\boldsymbol{F}_s$ and $\boldsymbol{C}_s$, are used to fit two GPs, namely $\boldsymbol{F}_s(\boldsymbol{x})$ and $\boldsymbol{C}_s(\boldsymbol{x})$, according to (2) and modelling:

$$\boldsymbol{f}_s(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}_s(\boldsymbol{x}), \boldsymbol{\sigma}_s^2(\boldsymbol{x})) \tag{6}$$

and

$$\boldsymbol{C}_s(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{p}_s(\boldsymbol{x}), \boldsymbol{q}_s^2(\boldsymbol{x})) \tag{7}$$

with $\boldsymbol{s} = \boldsymbol{1}, \ldots, \boldsymbol{S}$ and where $\boldsymbol{p}_s(\boldsymbol{x})$ and $\boldsymbol{q}_s(\boldsymbol{x})$ are again mean, and standard deviation, but different symbols are used to distinguish them from $\boldsymbol{\mu}_s(\boldsymbol{x})$ and $\boldsymbol{\sigma}_s(\boldsymbol{x})$.

3.2 Augmented Gaussian Process

An Augmented Gaussian Process (AGP) [21] is aimed at generating a single model over all the $\boldsymbol{f}_s(\boldsymbol{x})$ depending on the simplified model discrepancy measure, which is devoted to measuring the difference between two GPs approximating two different sources:

$$\boldsymbol{\eta}(\boldsymbol{F}_s, \boldsymbol{F}_{\bar{s}}, \boldsymbol{x}) = |\boldsymbol{\mu}_s(\boldsymbol{x}) - \boldsymbol{\mu}_{\bar{s}}(\boldsymbol{x})| \tag{8}$$

with $\boldsymbol{s}, \bar{\boldsymbol{s}} = \boldsymbol{1}, \ldots, \boldsymbol{S}$ and $\boldsymbol{s} \neq \bar{\boldsymbol{s}}$.

The value in (8) is a simplified discrepancy measure, adopted because it avoids estimating full covariance structures at every candidate location during acquisition optimization. More expressive alternatives exist, briefly discussed below, but were not adopted in MISO-wiLDCosts for this computational reason.

A significant measure is provided by the Wasserstein-2 distance. For two Gaussian distributions $\boldsymbol{\mu} = \boldsymbol{N}(\boldsymbol{m}_\mu, \boldsymbol{\Sigma}_\mu)$ and $\boldsymbol{\nu} = \boldsymbol{N}(\boldsymbol{m}_\nu, \boldsymbol{\Sigma}_\nu)$, the squared Wasserstein-2 distance admits a closed-form expression known as the Bures metric:

$$\boldsymbol{W}_2^2(\boldsymbol{\mu}, \boldsymbol{\nu}) = \| \boldsymbol{m}_\mu - \boldsymbol{m}_\nu \|^2 + \boldsymbol{B}\left(\boldsymbol{\Sigma}_\mu, \boldsymbol{\Sigma}_\nu\right)^2 \tag{9}$$

where $\mathbf{B}$ is the so-called Bures metric [43] between positive definite matrices [44],

$$\boldsymbol{B}\left(\boldsymbol{\Sigma}_\mu, \boldsymbol{\Sigma}_\nu\right)^2 \overset{\text{def}}{=} \boldsymbol{tr}(\boldsymbol{\Sigma}_\mu + \boldsymbol{\Sigma}_\nu - \boldsymbol{2}\left(\boldsymbol{\Sigma}_\mu^{\frac{1}{2}}\boldsymbol{\Sigma}_\nu\boldsymbol{\Sigma}_\mu^{\frac{1}{2}}\right)^{1/2}) \tag{10}$$

where $\boldsymbol{\Sigma}^{1/2}$ denotes the matrix square root. This formula follows from computing the transport cost of the optimal map between two Gaussians: the first term $\| \boldsymbol{m}_\mu - \boldsymbol{m}_\nu \|^2$ accounts for the displacement of means, while the Bures term captures the cost of transporting the covariance structure, which involves a geometric

mean of the two covariance matrices. When $\boldsymbol{\Sigma}_\mu$ and $\boldsymbol{\Sigma}_\nu$ commute, the expression simplifies to $\boldsymbol{B}\left(\boldsymbol{\Sigma}_\mu, \boldsymbol{\Sigma}_\nu\right)^2 = \| \boldsymbol{\Sigma}_\mu^{\frac{1}{2}} - \boldsymbol{\Sigma}_\nu^{\frac{1}{2}} \|_F^2$, where $\| \cdot \|_F$ is the Frobenius norm.

A pointwise approach involves considering the kernel difference, also known as the Difference of Gaussians (DoG). Given two Gaussian kernels with variances $\boldsymbol{\sigma}_1^2 < \boldsymbol{\sigma}_2^2$,

$$\boldsymbol{K}_{\sigma_1,\sigma_2}(\boldsymbol{x}) = \frac{\mathbf{1}}{(2\pi\sigma_1^2)^{n/2}}\mathbf{exp}(-\frac{\|\boldsymbol{x}\|^2}{2\sigma_1^2}) - \frac{\mathbf{1}}{(2\pi\sigma_2^2)^{n/2}}\mathbf{exp}(-\frac{\|\boldsymbol{x}\|^2}{2\sigma_2^2}) \tag{11}$$

obtained by subtracting the higher-variance Gaussian from the lower-variance one. This kernel acts as a band-pass filter: it suppresses both high-frequency noise (handled by the narrow kernel) and low-frequency background (handled by the wide kernel), retaining only intermediate-scale structure.

In MISO-wiLDCosts, we adopt the simplified difference in (8) and reconsider the role of variance in (9). The preferred source is defined as $\boldsymbol{f}_1(\boldsymbol{x})$, the source with the highest query cost, which, following the convention in Section 3.1 where sources are sorted by decreasing cost, corresponds to the most expensive and typically most accurate source. In the HPO experiments of this paper, $\boldsymbol{f}_1(\boldsymbol{x})$ coincides with $\boldsymbol{f}(\boldsymbol{x})$, i.e., evaluation on the full dataset. Fitting the AGP requires augmenting the set of function evaluations performed on the preferred source (i.e., the $\boldsymbol{F}_1$ set), with function evaluations performed on other sources whose function value is sufficiently close to the prediction provided by $\boldsymbol{F}_1(\boldsymbol{x})$. More formally, the set of augmenting locations, namely $\overline{\boldsymbol{F}}$, is given by

$$\overline{\boldsymbol{F}} = \{(\boldsymbol{x}, \boldsymbol{y}_s) \in \boldsymbol{F}_s, \boldsymbol{s} = \mathbf{2}, \ldots, \boldsymbol{S}: \boldsymbol{\eta}(\boldsymbol{F}_s, \boldsymbol{F}_1, \boldsymbol{x}) < \boldsymbol{m}\boldsymbol{\sigma}_1(\boldsymbol{x})\} \tag{12}$$

where $\boldsymbol{\eta}(\boldsymbol{F}_s, \boldsymbol{F}_1, \boldsymbol{x})$ is computed according to (8). The parameter m controls the discrepancy threshold for selecting augmenting observations. In the experiments, $\boldsymbol{m} = \mathbf{1}$, so an observation from source s is included when its discrepancy from the preferred-source prediction is smaller than one preferred-source predictive standard deviation.

The final set of inducing locations to use for fitting the AGP is given by $\boldsymbol{F}_b = \boldsymbol{F}_1 \cup \overline{\boldsymbol{F}}$, and the resulting AGP is denoted by $\boldsymbol{F}_b(\boldsymbol{x})$, such that:

$$\boldsymbol{F}_b(\boldsymbol{x}) \mid \widehat{\boldsymbol{F}} \sim \boldsymbol{N}(\widehat{\boldsymbol{\mu}}_b(\boldsymbol{x}), \widehat{\boldsymbol{\sigma}}_b^2(\boldsymbol{x})) \tag{13}$$

where $\widehat{\boldsymbol{F}}$ is the augmented dataset and the posterior mean $\widehat{\boldsymbol{\mu}}_b(\boldsymbol{x})$ and variance $\widehat{\boldsymbol{\sigma}}_b^2(\boldsymbol{x})$ are computed using the GP posterior equations in (2). The notation $\boldsymbol{\mu}_s(\boldsymbol{x})$ and $\boldsymbol{\sigma}_s(\boldsymbol{x})$ is reserved for the posterior mean and standard deviation of the source-specific GP $\boldsymbol{F}_s(\boldsymbol{x})$, whereas $\widehat{\boldsymbol{\mu}}_b(\boldsymbol{x})$ and $\widehat{\boldsymbol{\sigma}}_b^2(\boldsymbol{x})$ denote the corresponding quantities for the AGP $\boldsymbol{F}_b(\boldsymbol{x})$. Thus, Eq. (13) defines the AGP and not an individual source GP.

3.3 Selecting the Next Source-Location to Query

To select the next source and locations to query, we started from the acquisition function proposed in [21] according to the AGP model. However, we have significantly modified it to consider the location-dependent query cost of each source. The resulting acquisition function is based on the well-known Lower Confidence Bound (LCB), whose convergence proof under an appropriate scheduling of its technical parameter, $\boldsymbol{\beta}^t$, is given in [45]. The acquisition function proposed in this paper is defined as follows:

$$(\acute{s}, \acute{x}) = \boldsymbol{arg} \max_{\substack{\boldsymbol{s} = 1 \ldots \boldsymbol{S} \\ \boldsymbol{x} \in \boldsymbol{\Omega} \subset \mathcal{R}^d}} \frac{\widehat{\boldsymbol{y}}^+ - \left[\widehat{\boldsymbol{\mu}}_b(\boldsymbol{x}) - \sqrt{\boldsymbol{\beta}^t}\widehat{\boldsymbol{\sigma}}_b^2(\boldsymbol{x})\right]}{\mathbf{1} + \boldsymbol{c}_s^{\Delta}(\boldsymbol{x})\boldsymbol{\eta}(\widehat{\mathcal{F}}, \mathcal{F}_s, \boldsymbol{x})} \tag{14}$$

where $\widehat{\boldsymbol{y}}^{+}$ is the lowest function value of the inducing locations set $\widehat{\boldsymbol{F}}$, that is $\widehat{\boldsymbol{y}}^{+} = \mathbf{min}_{(\boldsymbol{x},\boldsymbol{y})\in\widehat{\boldsymbol{F}}}\{\boldsymbol{y}\}$, and $\boldsymbol{c}_{\boldsymbol{s}}^{\Delta}(\boldsymbol{x})$ is an estimation of the query cost for the source s.

Therefore, the numerator of (14) is the most optimistic improvement, with respect to $\widehat{\boldsymbol{y}}^{+}$, depending on the AGP's LCB, the denominator consists of two *source-and-location-dependent* penalization terms: $\boldsymbol{c}_{\boldsymbol{s}}^{\Delta}(\boldsymbol{x})$ and the model discrepancy between the AGP $\boldsymbol{F}_{\boldsymbol{b}}(\boldsymbol{x})$ and the source-specific GP $\boldsymbol{F}_{\boldsymbol{s}}(\boldsymbol{x})$.

The location-dependent query cost is modelled by the GP $\boldsymbol{C}_{\boldsymbol{s}}(\boldsymbol{x})$. A risk-averse attitude is adopted: rather than using the posterior mean of $\boldsymbol{c}_{\boldsymbol{s}}^{\Delta}(\boldsymbol{x})$, the denominator of (14) uses its upper confidence bound—the most pessimistic (highest) estimate of the cost to query $\boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{x})$. This conservative choice discourages the acquisition function from selecting source-location pairs whose cost is highly uncertain and potentially high. Formally, the cost term is taken as $\mathbf{max}\{\mathbf{0}, \boldsymbol{p}_{\boldsymbol{s}}(\boldsymbol{x}) + \boldsymbol{q}_{\boldsymbol{s}}(\boldsymbol{x})\}$ to handle negative values that may arise from GP approximation. Consequently, when two source-location pairs provide similar expected optimization benefit, the acquisition function favors the one with lower predicted cost uncertainty, leading to more reliable cost-efficient exploration.

Finally, as reported in [21], there is the possibility that solving (14) could lead to choosing a pair $\left(\boldsymbol{s}^{\mathbf{0}}, \boldsymbol{x}^{\mathbf{0}}\right)$ whose location $\boldsymbol{x}^{\mathbf{0}}$ is very close to a previously evaluated location on the source $\boldsymbol{s}^{\mathbf{0}}$, leading to the ill-conditioning of the matrix $\left[\boldsymbol{K} + \boldsymbol{\lambda}^{\mathbf{2}}\boldsymbol{I}\right]$ in (2) and, consequently, to the impossibility to update $\boldsymbol{\mu}_{\boldsymbol{s}}(\boldsymbol{x})$ and $\boldsymbol{\sigma}_{\boldsymbol{s}}^{\mathbf{2}}(\boldsymbol{x})$. The correction proposed to avoid this undesired behaviour is:

Correction: If $\exists(\mathbf{x}, \mathbf{y}) \in \mathbf{F}_{\mathbf{s}^0}: \|\mathbf{x} - \mathbf{x}^{\mathbf{0}}\| < \boldsymbol{\delta}$ then set $\mathbf{s}^{\mathbf{0}} \leftarrow \mathbf{1}$ and choose $\mathbf{x}^{\mathbf{0}}$ as follows:

$$\mathbf{x}^{\mathbf{0}} \leftarrow \boldsymbol{argmax}_{\boldsymbol{x}\in\boldsymbol{\Omega}}\boldsymbol{\sigma}_{\mathbf{1}}(\boldsymbol{x}) \tag{15}$$

The idea is to choose an alternative $\mathbf{x}^{\mathbf{0}}$ by "investing" the available budget in exploration to improve the AGP at the next iteration by querying the most expensive source. Indeed, as stated in [45], selecting the location associated with the highest prediction uncertainty is a good strategy for function learning (aka function approximation), whose goal is to efficiently explore the search space to obtain an accurate approximation of $\boldsymbol{f}(\boldsymbol{x})$ within a limited number of queries. It is important to note that the convergence guarantees available for standard confidence-bound acquisition functions [45] do not automatically extend to MISO-wiLDCosts once the discrepancy and cost-penalization terms in (14) are introduced; a formal regret analysis for this setting is left for future work. Algorithm 1 summarizes the full MISO-wiLDCosts procedure. Algorithm 1 summarizes the full MISO-wiLDCosts procedure.

Algorithm 1: MISO-wiLDCosts

Input: Search space Ω, sources $\{f_s\}_{s=1:S}$, budget τ, initial design size n_0 per source, parameter m, κ
Output: Best observed configuration x^* and its value y_1^* on the primary source

1. Initialization: For each source $s = 1, \ldots, S$, sample n_0 locations $\{x^i\}_{i=1}^{n_0}$ via Latin Hypercube Sampling; query y_s^i and record cost z_s^i. Set $D_s \leftarrow \{x^i, y_s^i, z_s^i\}_{i=1}^{n_0}$, cumulated cost $\tau_{\text{curr}} \leftarrow \sum_{s,i} z_s^i$.

2. Fit initial GPs: For each source s, fit $f_s(x) \sim \mathcal{N}(\mu_s(x), \sigma_s^2(x))$ on F_s, and $C_s(x) \sim \mathcal{N}(p_s(x), q_s^2(x))$ on C_s via MLE (Matérn 3/2 kernel).

3. Build AGP: For every non-preferred source $s = 2, \ldots, S$ compute the discrepancy $\eta(F_s, F_1, x)$ using Eq. (8). Construct the augmenting set $\overline{F}$ using Eq. (12), and define the augmented dataset $\widehat{F} = F_1 \cup \overline{F}$. Fit the AGP $F_b(x) \sim N(\widehat{\mu}_b(x), \widehat{\sigma}_b^2(x))$ on $\widehat{F}$ using Eq. (2).

4. While $\tau_{\text{curr}} < \tau$ **do:**

(Continued)

Algorithm 1 (continued)

a. For every source $s = 1, \ldots, S$ and candidate location $x \in \Omega$, compute:

$$A_s(x) = \frac{\widehat{\boldsymbol{y}}^{+} - \left[\widehat{\boldsymbol{\mu}}_{\boldsymbol{b}}(\boldsymbol{x}) - \sqrt{\boldsymbol{\beta}^{t}}\widehat{\boldsymbol{\sigma}}_{\boldsymbol{b}}^{2}(\boldsymbol{x})\right]}{\mathbf{1} + \boldsymbol{c}_{\boldsymbol{s}}^{\Delta}(\boldsymbol{x})\,\boldsymbol{\eta}\left(\widehat{\mathcal{F}}, \mathcal{F}_{\boldsymbol{s}}, \boldsymbol{x}\right)}\mathbf{arg}$$

(Eq. (14) in the paper)

b. Select next query: $(\widehat{s}, \widehat{x}) = \arg\max_{s,x} A_s(x)$

c. Query source $\tilde{s}$ at $\tilde{x}$: observe $y_{\tilde{s}}$, record cost $z_{\tilde{s}}$

d. Update $D_{\tilde{s}} \leftarrow D_{\tilde{s}} \cup \{(\tilde{x}, y_{\tilde{s}}, z_{\tilde{s}})\}$; update $\tau_{\text{curr}} \leftarrow \tau_{\text{curr}} + z_{\tilde{s}}$

e. Re-fit GPs for source $\tilde{s}$: update $f_{\tilde{s}}(x)$ and $C_{\tilde{s}}(x)$

f. Rebuild AGP (Step 3) using all updated $\{D_s\}_{s=1:S}$

5. Return $x^* = \arg\min_{x \in \bigcup_s F_s^{(1)}} \tilde{\mu}(x)$, where $F_s^{(1)}$ are evaluations on the primary source $s = 1$

4 Experimental Setting

4.1 Task Description

To evaluate MISO-wiLDCosts, we considered a core application, that is, the HPO of an ML algorithm. This task has been widely addressed via MISO/multifidelity optimization, such as in [17,18,21–23], as well as cost-aware optimization [30–32]. Three binary classification datasets have been considered, different in size and number of features:

SPLICE(3)[1]–related to primate splice-junction gene sequences with associated imperfect domain theory, and used in [30]. This dataset consists of 3175 instances, 60 (numeric) features, plus the (binary) class label.

SVMGUIDE(1)[2]–related to an astroparticle application from Jan Conrad of Uppsala University, Sweden [46]. This dataset consists of 7089 instances, 4 (numeric) features, plus the (binary) class label.

MAGIC GAMMA TELESCOPE[3]–generated through a Monte Carlo simulation software, namely Corsika, described in [47]. The dataset has been used in [21] and consists of 19,020 instances, 10 (numeric) features plus the (binary) class label.

All the features of the three datasets have been preliminarily scaled in $[0, 1]$.

The following ML classification algorithms, and associated hyperparameters, have been considered for the HPO task on the three datasets:

Support Vector Machine classifier (C-SVC) with Radial Basis Function (RBF) kernel [48]. The hyperparameters to optimize are: the SVC's regularization term $C \in [10^{-2}, 10^{2}]$ and $\gamma \in [10^{-4}, 10^{4}]$ of the RBF kernel (i.e., $k(a, \acute{a}) = e^{\frac{-\|a-\acute{a}\|^2}{2\gamma^2}}$, with $a \neq \acute{a}$ two instances within the dataset).

Random Forest (RF) classifier [49]. The hyperparameters to optimize are the number of decision trees in the forest, $n_{trees} = \{300, \ldots, 700\}$, and the number of features subsampled to generate every tree, $m_{try} \in \left[\lfloor 0.25 \times m_{feat} \rfloor, \lfloor 0.75 \times m_{feat} \rfloor\right]$, with m_{feat} the number of features in the original dataset. The symbol $\lfloor . \rfloor$ represents the rounding operation to the closest integer value.

[1] https://www.openml.org/d/1579
[2] https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/binary.html#svmguide1
[3] https://archive.ics.uci.edu/ml/datasets/magic+gamma+telescope

The goal is to identify, for each dataset and for each ML classifier, the hyperparameter values minimizing the mis-classification error (mce), computed:

- on 10-fold cross-validation (mce-FCV) for the CSVC
- and Out-Of-Bag (mce-OOB) for the RF classifier while keeping the cumulated query cost as small as possible

Figs. 1 and 2 show, respectively, for the CSVC and the RF classifier, the query cost and the mis-classification error of 1000 hyperparameter configurations, randomly sampled via LHS, for each one of the three datasets. The blue line can be assimilated to a Pareto frontier, minimizing both mce and query cost, and it is used to make more evident the relation between the optimal mce and its cost.

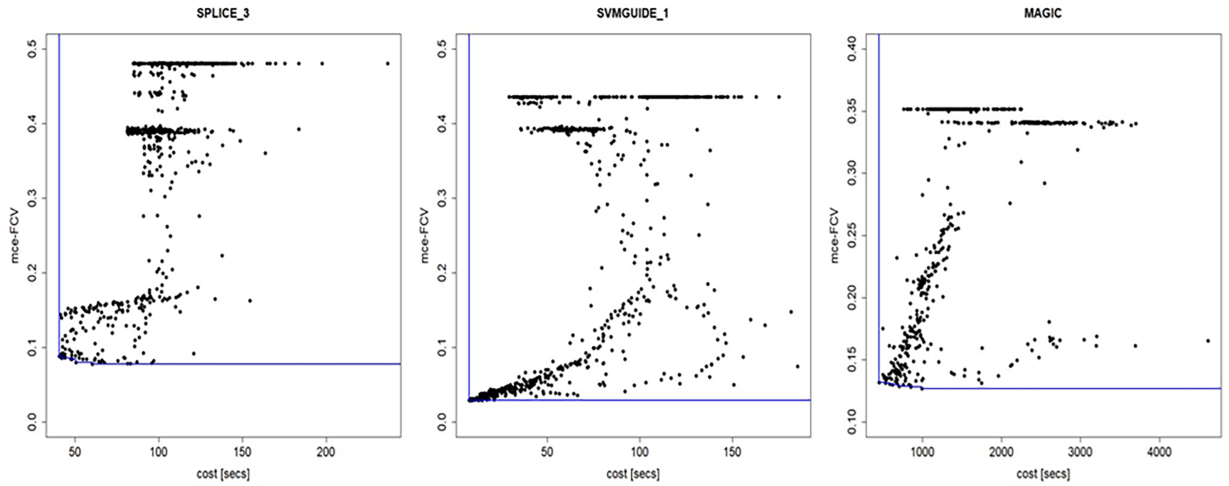

Figure 1: Query cost and mce-FCV of 1000 randomly sampled C-SVC's hyperparameter configurations on three datasets: SPLICE(3) (left), SVMGUIDE(1) (middle), and MAGIC (right). The blue line can be assimilated to a Pareto frontier minimizing both mce and cost. In this case, the minimum mce should also be cheap.

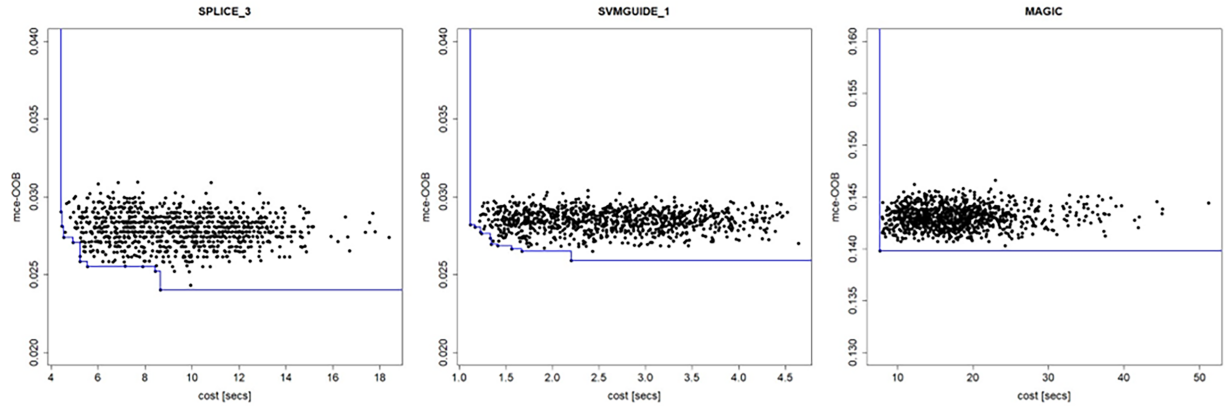

Figure 2: Query cost and mce-OOB of 1000 randomly sampled RF's hyperparameter configurations on three datasets: SPLICE(3) (left), SVMGUIDE(1) (middle), and MAGIC (right). The blue line can be assimilated to a Pareto frontier minimizing both mce and cost. For the SPLICE(3) and SVMGUIDE(1) datasets, the minimum mce is not associated with the cheapest RF's hyperparameters.

First, when the two figures are compared, C-SVC's and RF's hyperparameter configurations have completely different mce values and query costs. This is mainly due to both the high computational cost for training an SVM classifier (i.e., number of instances to the power of three) and differences in the computation of mce (i.e., on 10 FCV for C-SVC and OOB for RF). Then, C-SVC's mce values vary in a larger range than RF's ones. Therefore, the two classification algorithms can be considered two different representative cases. With respect to C-SVC (Fig. 1), the sampled hyperparameter configurations show that (*i*) the optimum mce should be "cheap"—especially for the SVMGUIDE(1) dataset—but (*ii*) it could be difficult to reach, according to the small number of configurations around its minimum observed value, for all three datasets. On the contrary, the minimum mce observed for the RF classifier is not associated with the cheapest hyperparameter configurations in the case of SPLICE(3) and SVMGUIDE(1) datasets (Fig. 2). Therefore, a lower mce can be achieved by more expensive RF classifiers, on these two datasets (even if it would not be significantly different from the average).

4.2 Sources Setup

In MISO and multi-fidelity optimization for HPO, information sources are typically associated with small portions of the large original dataset. A similar idea is followed in this paper: each one of the three datasets was divided into 10 stratified subsets, then reaggregated to generate the following five sources:

f_1 (x) = f(x), mce-FCV and mce-OOB on the entire dataset, respectively for C-SVC and RF;

f_2 (x), mce-FCV and mce-OOB on the 40% of the original dataset (merging the first 4 subsets);

f_3 (x), mce-FCV and mce-OOB on the 30% of the original dataset (merging subsets 5 to 7);

f_4 (x), mce-FCV and mce-OOB on the 20% of the original dataset (merging subsets 8 and 9);

f_5 (x), mce-FCV and mce-OOB on the 10% of the original dataset (just the subset 10)

According to this experimental setup, $f_2(x)$ to $f_5(x)$ rely on subsets that do not overlap. Although this could imply a hierarchical organization of the sources, we assume that we do not anything about the composition and nature of the sources; they are completely black boxes for the study.

4.3 MISO-wiLDCosts Setup

A set of 5 initial locations (i.e., hyperparameter values) sampled via Latin Hypercube Sampling (LHS), on each source, is used to initialize MISO-wiLDCosts. Ten independent runs are performed for each dataset and for each classification algorithm to mitigate the effect of random initialization.

The initialization procedures of CArBO cost-cooling and MISO-wiLDCosts are different. CArBO cost-cooling starts with five configurations evaluated on the full-data source, whereas MISO-wiLDCosts starts with five configurations on each of the five sources, resulting in 25 initial source evaluations. Consequently, the comparison is not matched in terms of the number of initial evaluations. The primary comparison in this study is based on cumulative query cost, because evaluations on the different sources have different computational costs. However, the initial cumulative costs cannot be reported retrospectively because the run-level initialization costs from the original experiments are not available. We therefore acknowledge the unequal initialization as a limitation of the comparison.

At a generic iteration of MISO-wiLDCosts, all the GP models—$F_s(x)$, $C_s(x)$ and $\widehat{F}_s(x)$—are updated conditioned to the function values and costs observed so far. Then, the next $(\acute{s}, \acute{x})$ to query is selected according to (14), and in case (15). Model updating and evaluation of the next selected source location are iterated until fifty function evaluations are performed.

As the final solution of each run, MISO-wiLDCosts returns the hyperparameter values, x^+, associated with the $\widehat{y}^+$ obtained at the end of the process, that is $x^+ : (x^+, \widehat{y}^+) \in \widehat{F}$. This requires computing, one last time, the set of the AGP's inducing locations, $\widehat{F}$. Finally, since x^+ could be a location queried on a source different from $f_1(x)$, a further evaluation is performed to replace $y\mathrm{b}^+$ with the $f_1(x^+)$. Although this situation has been considered in implementing MISO-wiLDCosts, it never occurred in the experiments performed.

4.4 Baseline

Since no prior work has simultaneously addressed MISO and cost-aware optimization, baseline selection requires careful justification. MISO-wiLDCosts sits at the intersection of two research directions: multi-source optimization and cost-aware BO. The relevant baselines must therefore cover both dimensions. For cost-aware single-source BO, CArBO [30] represents the state of the art and is the appropriate comparison: adding weaker cost-aware alternatives such as EI-per-cost [31] would only widen the performance gap in favor of MISO-wiLDCosts without providing additional scientific insight. For multi-source optimization without cost-awareness, the natural baseline is MISO-AGP [21], the method that MISO-wiLDCosts directly extends. Rather than replicating published MISO-AGP experiments, we note that Ref. [21] establishes the performance of AGP-based multi-source fusion on HPO tasks without cost-awareness; the comparison between MISO-wiLDCosts and CArBO in this paper thus implicitly spans both dimensions: CArBO represents the cost-aware single-source upper bound, and Ref. [21] represents the multi-source cost-unaware reference point. A unified experimental comparison including all three methods on identical tasks remains a valuable direction for future work. It is important to note that CArBO also implements a procedure to obtain a cost-effective initial design, which could also be included in MISOwilDCosts. In this paper, just the cost-cooling of CArBO has been considered, while the initial designs are those sampled via LHS in the MISO-wiLDCosts experiments. Cost-cooling requires defining the overall budget in terms of the maximum cumulated query cost. We defined it to cover approximately the fifty function evaluations performed by MISO-wiLDCosts, leading to:

HPO of C-SVC: 1 h for SPLICE(3) and SVMGUIDE(1), and 3 h for MAGIC.

HPO of RF: 15 min for SPLICE(3), 10 min for SVMGUIDE(1), and 1 h and a half for MAGIC.

Just for a fair comparison, also for CArBO cost-cooling, the optimization process was stopped at fifty evaluations, even if some residual budget was available.

4.5 Computational Setting

MISO-wiLDCosts is developed in R. Experiments were run on a Microsoft Azure virtual machine, H8 (High Performance Computing family) Standard with 8 vCPUs, 56 GB of memory, Ubuntu 16.04.6 LTS.

5 Results

The main results of our experiments are summarized in Table 1, where the suffixes "wild" and "cooling" are used to distinguish between MISO-wiLDCosts and CArBO cost-cooling, respectively. Unsurprisingly, cumulated query cost is significantly lower for MISO-wiLDCosts, according to Wilcoxon's nonparametric test for paired samples ($\boldsymbol{p_{value}} < \mathbf{0.001}$). The most relevant outcomes related to mce are:

- With respect to C-SVC, mce-wild is higher than mce-cooling, for SPLICE(3) ($\boldsymbol{p_{value}} < \mathbf{0.0001}$) and SVMGUIDE(1) datasets ($\boldsymbol{p_{value}} = \mathbf{0.01}$);
- mce-wild is equal to mce-cooling in the case of HPO of C-SVC on the MAGIC dataset (Tables 1, 2). The reduction in query cost is smaller than in the other C-SVC cases but still favors cost-cooling (Table 1).

- As far as HPO of RF is concerned, mce-wild and mce-cooling are basically the same, but MISOwiLDcosts used less than 40% of the cumulated cost required by CArBO cost-cooling, less than 20% for the SPLICE(3) and SVMGUIDE(1) datasets.

These differences are more evident in Table 2, where "delta mce" is the difference between mce-wild and mce-cooling, and "%cost" is given by $\mathbf{100} \times \frac{cost-wild}{cost-cooling}$.

It should be noted that the comparison is not fully symmetric: our implementation uses LHS-based initial designs rather than CArBO's cost-effective initial design, which may modestly favor CArBO in early iterations.

Table 1: HPO results on three classification datasets and two ML classification algorithms: mce and cumulated query cost for 10 runs of MISO-wiLDCosts and CArBO cost-cooling.

Classifier	Dataset	mce-wild	mce-cooling	cost-wild [s]	cost-cooling [s]
C-SVC	SPLICE(3)	0.168 ± 0.062	0.080 ± 0.002	744.607 ± 91.760	3646.908 ± 30.841
	SVMGUIDE(1)	0.037 ± 0.014	0.029 ± 0.000	572.063 ± 96.167	3263.303 ± 385.737
	MAGIC	0.172 ± 0.000	0.172 ± 0.000	11,050.060 ± 522.669	11,575.700 ± 699.917
RF	SPLICE(3)	0.025 ± 0.003	0.025 ± 0.000	78.624 ± 7.375	456.572 ± 42.073
	SVMGUIDE(1)	0.026 ± 0.003	0.027 ± 0.000	25.230 ± 1.490	133.070 ± 8.460
	MAGIC	0.141 ± 0.000	0.141 ± 0.000	492.327 ± 39.704	1360.064 ± 349.545

Table 2: Comparison between MISO-wiLDCosts and CArBO cost-cooling.

Classifier	Dataset	delta mce	%cost
C-SVC	SPLICE (3)	0.088 ± 0.063	20.43% ± 2.60%
	SVMGUIDE(1)	0.008 ± 0.014	17.68% ± 2.99%
	MAGIC	0.000 ± 0.000	95.64% ± 5.04%
RF	SPLICE(3)	0.000 ± 0.003	17.26% ± 1.31%
	SVMGUIDE(1)	−0.001 ± 0.003	19.04% ± 1.78%
	MAGIC	0.001 ± 0.000	38.65% ± 11.12%

Finally, just for illustrative purposes, Fig. 3 shows the query cost incurred at each iteration of MISOwiLDCosts and CArBO cost-cooling, separately (solid vs. dashed lines): lines and shaded area are, respectively, the mean and standard deviation computed on the 10 independent runs. For reasons of space and ease of viewing, the figure refers to HPO of RF, but the behaviour is analogous for HPO of C-SVC.

It is important to note that just five initial hyperparameter configurations are used to initialize CArBO cost-cooling, while five for each source (i.e., twenty-five) are used to initialize MISO-wiLDCosts.

After the initial design, MISO-wiLDCosts mainly used "cheap" sources in the case of SPLICE(3) and SVMGUIDE(1) datasets, while it used all the sources, including $\boldsymbol{f}_1(\boldsymbol{x})$, in the case of the MAGIC dataset. This means that, in the case of SPLICE(3) and SVMGUIDE(1): (*i*) sources that were *discrepant* from $\boldsymbol{f}_1(\boldsymbol{x})$ had contributed in providing inducing locations of the AGP and/or (*ii*) any early convergence towards a specific source-location pair, $(\acute{\boldsymbol{s}}, \acute{\boldsymbol{x}})$, did not occur, thus correction (12) was not (frequently) used.

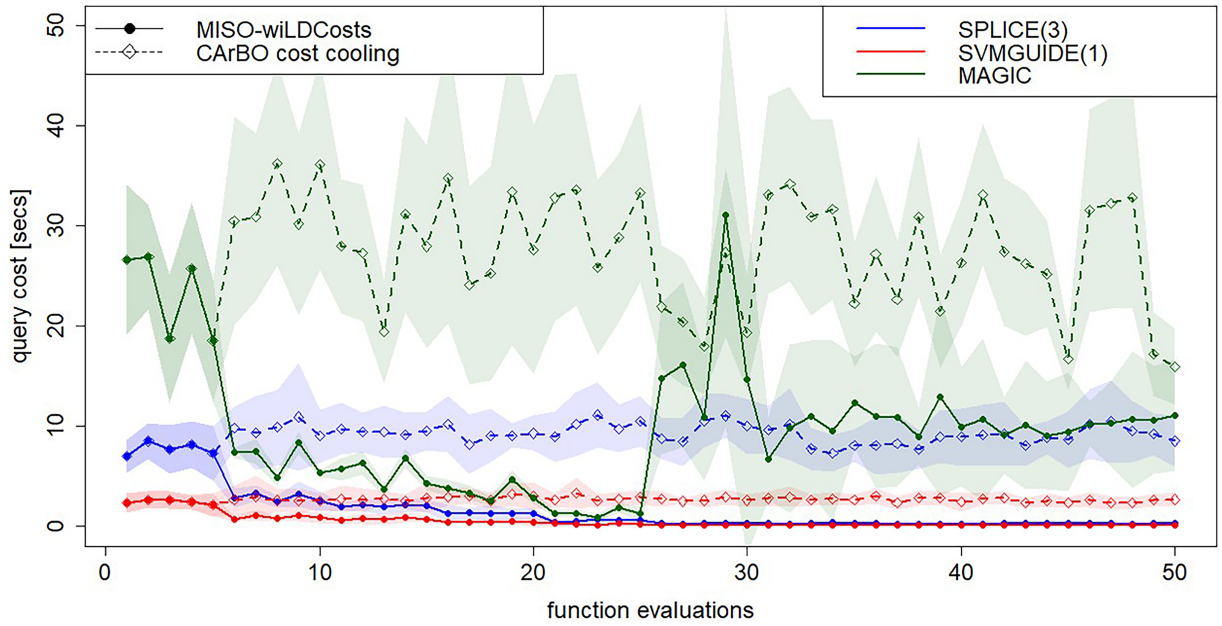

Figure 3: RF classifier HPO on three datasets: query cost, by function evaluation, for MISO-wiLDCosts and CArBO cost-cooling. Lines and shaded areas are, respectively, the mean and standard deviation on 10 runs. Initial design for CArBO cost-cooling consists of five hyperparameter configurations sampled via LHS, while for MISOwiLDCosts, it consists of five evaluations: five hyperparameter configurations on each one of the five sources.

6 Conclusions

This paper shows that unifying MISO and cost-aware BO within a single framework is feasible. MISO-wiLDCosts is, to the best of our knowledge, the first MISO approach that also accounts for location-dependent costs within each source. Its practical value is demonstrated on an HPO task, where MISO-wiLDCosts achieves a favorable trade-off between cumulated query cost and classification accuracy compared to single-source cost-aware BO, with substantially lower cost at the price of a small, dataset-dependent increase in misclassification error in some settings.

Limitations: A limiting assumption is that different sources, despite differing observational cost and quality, live in the same space. When the sources are different, e.g., a continuous or a grid vs. the output of a FEM in the fluid dynamics solver, or a physics-inspired machine learning model, or a physical prototype, the acquisition formula is still valid, but computing the distance between Gaussians requires different models. A further limitation concerns the absence of ablation studies isolating the contribution of individual components. In particular, comparing MISO-wiLDCosts against a variant using standalone per-source GP fusion without the AGP discrepancy penalty would clarify the specific benefit of the augmentation mechanism. The current evaluation is limited to HPO tasks, where both source fidelity and query cost are inherently entangled with the underlying ML algorithm and dataset. Controlled experiments on synthetic benchmark functions with explicitly constructed location-dependent fidelity and cost structures would allow a cleaner assessment of the proposed acquisition function's behavior independently of the AutoML setting. The simplified discrepancy measure adopted in (8) ignores predictive uncertainty, relying solely on posterior means. An empirical sensitivity analysis comparing this measure against uncertainty-aware alternatives such as the Wasserstein-2/Bures distance would provide stronger justification for this design choice and is left for future work. Another limitation concerns the initialization budget. MISO-wiLDCosts uses five initial configurations per source, while CArBO uses five configurations on the full-data source. Because

the run-level initialization costs are not available, we could not report the corresponding numerical initial cumulative costs or perform a retrospective sensitivity analysis with matched initialization. A cost-matched or evaluation-matched initialization study is therefore an important direction for future work.

Perspectives: The framework of multi-source Bayesian optimization can be extended to incorporate alternative divergence measures beyond f-divergences, such as Wasserstein distance, which provides a natural metric for quantifying movement costs when source domains share the same space [50]. The growing interest in Bayesian optimization with movement costs reflects practical constraints where the cost of transitioning between design points depends on both current and previous evaluations, fundamentally differing from conventional cost-aware approaches [37,51]. This creates a dual trade-off between exploration-exploitation and query costs that includes movement penalties, which can be more effectively addressed through lookahead acquisition function variants [51]. Recent developments have expanded this perspective by considering cost-varying variable subsets [34] and intervention costs in causal systems [52], suggesting that when source spaces differ structurally, such as continuous domains versus finite element method outputs.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: conceptualization, Antonio Candelieri and Francesco Archetti; methodology, Antonio Candelieri and Francesco Archetti; software, Antonio Candelieri; validation, Antonio Candelieri, Francesco Archetti and Iman Seyedi; formal analysis, Antonio Candelieri; investigation, Antonio Candelieri, Francesco Archetti and Iman Seyedi; resources, Antonio Candelieri, Francesco Archetti and Iman Seyedi; data curation, Antonio Candelieri and Francesco Archetti; writing—original draft preparation, Antonio Candelieri and Francesco Archetti; writing—review and editing, Antonio Candelieri, Francesco Archetti and Iman Seyedi; visualization, Antonio Candelieri and Iman Seyedi; supervision, Antonio Candelieri and Francesco Archetti; project administration, Antonio Candelieri and Francesco Archetti; funding acquisition, Antonio Candelieri and Francesco Archetti. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the Corresponding Author, [antonio.candelieri@unimib.it], upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Archetti F, Candelieri A. Bayesian optimization and data science. Cham, Switzerland: Springer International Publishing; 2019.
2. Frazier PI. Bayesian optimization. In: Recent advances in optimization and modeling of contemporary problems. Catonsville, MD, USA: INFORMS; 2018. p. 255–78.
3. Shahriari B, Swersky K, Wang Z, Adams RP, de Freitas N. Taking the human out of the loop: a review of Bayesian optimization. Proc IEEE. 2016;104(1):148–75. doi:10.1109/jproc.2015.2494218.
4. Candelieri A, Archetti F. Global optimization in machine learning: the design of a predictive analytics application. Soft Comput. 2019;23(9):2969–77. doi:10.1007/s00500-018-3597-8.
5. Hutter F, Kotthoff L, Vanschoren J. Automated machine learning: methods, systems, challenges. Cham, Switzerland: Springer International Publishing; 2019.
6. Elsken T, Metzen JH, Hutter F. Neural architecture search: a survey. J Mach Learn Res. 2019;20(55):1–21.

7. Sha D, Ozbay K, Ding Y. Applying Bayesian optimization for calibration of transportation simulation models. Transp Res Rec J Transp Res Board. 2020;2674(10):215–28. doi:10.1177/0361198120936252.
8. Candelieri A, Perego R, Archetti F. Bayesian optimization of pump operations in water distribution systems. J Glob Optim. 2018;71(1):213–35. doi:10.1007/s10898-018-0641-2.
9. Strubell E, Ganesh A, McCallum A. Energy and policy considerations for modern deep learning research. Proc AAAI Conf Artif Intell. 2020;34(9):13693–6. doi:10.1609/aaai.v34i09.7123.
10. Kennedy M, O'Hagan A. Predicting the output from a complex computer code when fast approximations are available. Biometrika. 2000;87(1):1–13. doi:10.1093/biomet/87.1.1.
11. Chaudhuri A, Marques AN, Lam R, Willcox KE. Reusing information for multifidelity active learning in reliability-based design optimization. In: Proceedings of the AIAA Scitech 2019 Forum; 2019 Jan 7–11; San Diego, CA, USA.
12. Kandasamy K, Dasarathy G, Oliva J, Schneider J, Póczos B. Multi-fidelity Gaussian process bandit optimisation. J Artif Intell Res. 2019;66:151–96. doi:10.1613/jair.1.11288.
13. Marques A, Lam R, Willcox K. Contour location via entropy reduction leveraging multiple information sources. In: Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018); 2018 Dec 2–8; Montréal, QC, Canada.
14. Candelieri A, Ponti A, Archetti F. Multiple information source Bayesian optimization. Cham, Switzerland: Springer Nature Switzerland; 2025.
15. March A, Willcox K. Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. AIAA J. 2012;50(5):1079–89. doi:10.2514/1.j051125.
16. Lam R, Allaire DL, Willcox KE. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In: Proceedings of the 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference; 2015 Jan 5–9; Kissimmee, FL, USA.
17. Poloczek M, Wang J, Frazier P. Multi-information source optimization. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017 Dec 4–9; Long Beach, CA, USA.
18. Ghoreishi SF, Allaire D. Multi-information source constrained Bayesian optimization. Struct Multidiscip Optim. 2019;59(3):977–91. doi:10.1007/s00158-018-2115-z.
19. Winkler RL. Combining probability distributions from dependent information sources. Manag Sci. 1981; 27(4):479–88. doi:10.1287/mnsc.27.4.479.
20. Ji S, Tan Y, Saravirta T, Yang Z, Liu Y, Vasankari L, et al. Emerging trends in federated learning: from model fusion to federated X learning. Int J Mach Learn Cybern. 2024;15(9):3769–90. doi:10.1007/s13042-024-02119-1.
21. Candelieri A, Perego R, Archetti F. Green machine learning via augmented Gaussian processes and multi-information source optimization. Soft Comput. 2021;25(19):12591–603. doi:10.1007/s00500-021-05684-7.
22. Swersky K, Snoek J, Adams RP. Multi-task Bayesian optimization. In: Proceedings of the Advances in Neural Information Processing Systems 26 (NIPS 2013); 2013 Dec 5–8; Lake Tahoe, NV, USA.
23. Klein A, Falkner S, Bartels S, Hennig P, Hutter F. Fast Bayesian optimization of machine learning hyperparameters on large datasets. Artif Intell Stat. 2017;54(2):528–36.
24. Oliva D, Martins MSR, Hinojosa S, Elaziz MA, dos Santos PV, da Cruz G, et al. A hyper-heuristic guided by a probabilistic graphical model for single-objective real-parameter optimization. Int J Mach Learn Cybern. 2022;13(12):3743–72. doi:10.1007/s13042-022-01623-6.
25. Park K, Molina Concha D, Lee HR, Lee T, Lee CG. Reward design in multi-agent systems using successor features and multi-information source Bayesian optimization. Int J Mach Learn Cybern. 2025;16(9):6249–70. doi:10.1007/s13042-025-02622-z.
26. Zhang Z, Zhang Z, Chen Z. Multiple-source localization from a single-snapshot observation using graph Bayesian optimization. Proc AAAI Conf Artif Intell. 2024;38(20):22538–46. doi:10.1609/aaai.v38i20.30262.
27. Zhou Z, Wang Y, Zhai Q, Yan J. Multi-source localization based on graph representation learning and Bayesian optimization. IEEE Trans Netw Sci Eng. 2026;13(3):4815–32. doi:10.1109/tnse.2025.3644931.
28. Mikkola P, Martinelli J, Filstroff L, Kaski S. Multi-fidelity Bayesian optimization with unreliable information sources. Int Conf Artif Intell Stat. 2023;206:7425–54.

29. Jacobs L, Ali Javidian M. Extending multi-source Bayesian optimization with causality principles. arXiv:2602.14791. 2026.
30. Lee EH, Perrone V, Archambeau C, Seeger M. Cost-aware Bayesian optimization. arXiv:2003.10870. 2020.
31. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012); 2012 Dec 3–6; Lake Tahoe, NV, USA.
32. Paria B, Neiswanger W, Ghods R, Schneider J, Póczos B. Cost-aware Bayesian optimization via information directed sampling. In: Proceedings of the Adaptive Experimental Design and Active Learning in the Real World Workshop at ICML; 2020 Jul 18; Virtual.
33. Russo D, Van Roy B. Learning to optimize via information-directed sampling. Adv Neural Inf Process Syst. 2014;27(1):1–9.
34. Tay S, Foo CS, Urano D, Leong R, Low BKH. Bayesian optimization with cost-varying variable subsets. Adv Neural Inf Process Syst. 2023;36:3008–31.
35. Liu P, Wang H, Qiyu W. Bayesian optimization with switching cost: regret analysis and lookahead variants. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence; 2023 Aug 19–25; Macao, China.
36. Kim H, Liu C, Chen Y. Bayesian optimization with inexact acquisition: is random grid search sufficient? arXiv:2506.11831. 2025.
37. Chen Q, Al Kontar R. The traveling bandit: a framework for Bayesian optimization with movement costs. Technometrics. 2026;68(2):239–50. doi:10.1080/00401706.2025.2573228.
38. Candelieri A, Archetti F, Seyedi I, Ponti A. Resource allocation via Bayesian optimization in Wasserstein spaces vs. semi-bandit feedback. Big Data Cogn Comput. 2026;10(7):206. doi:10.3390/bdcc10070206.
39. Jo S, Yoon S, Na HS, Kweon SJ. Multi-period large-scale electric vehicle charging station deployment under latent charging demand: a Bayesian optimization-based large neighborhood search approach. Expert Syst Appl. 2026;323(16):132054. doi:10.1016/j.eswa.2026.132054.
40. Pourmohamad T. Surrogates: Gaussian process modeling, design, and optimization for the applied sciences. Technometrics. 2021;63(1):144–5. doi:10.1080/00401706.2020.1865008.
41. Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Cambridge, MA, USA: The MIT Press; 2005.
42. Seyedi I, Candelieri A, Archetti F. Distributionally robust Bayesian optimization via sinkhorn-based Wasserstein barycenter. Mach Learn Knowl Extr. 2025;7(3):90. doi:10.3390/make7030090.
43. Bures D. An extension of Kakutani's theorem on infinite product measures to the tensor product of semifinite w*-algebras. Trans Am Math Soc. 1969;135:199–212. doi:10.2307/1995012.
44. Forrester PJ, Kieburg M. Relating the bures measure to the cauchy two-matrix model. Commun Math Phys. 2016;342(1):151–87. doi:10.1007/s00220-015-2435-4.
45. Srinivas N, Krause A, Kakade SM, Seeger MW. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Trans Inform Theory. 2012;58(5):3250–65. doi:10.1109/tit.2011.2182033.
46. Hsu CW, Chang CC, Lin CJ. A practical guide to support vector classification. 2003 [cited 2025 Jan 1]. Available from: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.
47. Heck D, Knapp J, Capdevielle JN, Schatz G, Thouw T. CORSIKA: a Monte Carlo code to simulate extensive air showers. Eggenstein-Leopoldshafen, Germany: Forschungszentrum Karlsruhe; 1998. Report No.: FZKA 6019.
48. Schölkopf B, Smola AJ. Learning with kernels: support vector machines, regularization, optimization, and beyond. Cambridge, MA, USA: MIT Press; 2002.
49. Goel E, Abhilasha E. Random forest: a review. Int J Adv Res Comput Sci Softw Eng. 2017;7(1):251–7. doi:10.23956/ijarcsse/v7i1/01113.
50. Seyedi I, Candelieri A, Messina E, Archetti F. Wasserstein distributionally robust optimization for chance constrained facility location under uncertain demand. Mathematics. 2025;13(13):2144. doi:10.3390/math13132144.
51. Ramesh SS, Sessa PG, Krause A, Bogunovic I. Movement penalized Bayesian optimization with application to wind energy systems. Adv Neural Inf Process Syst. 2022;35:27036–48. doi:10.52202/068431-1960.
52. Aglietti V, Lu X, Paleyes A, González J. Causal Bayesian optimization. Int Conf Artif Intell Stat. 2020;108:3155–64.