

Bayesian optimization with executable digital twins: Fast controller tuning with multi-source information

Giacomo Delcaro ¹, Simone Gabrielli *, Simone Formentin

Electronics, Computer Science and Bioengineering Department (DEIB), Politecnico di Milano, Milano, Italy

ARTICLE INFO

Keywords:

Bayesian optimization
Gaussian process
Multi-output Gaussian process
Acquisition function
Digital twin
Twin-in-the-loop
Controller tuning
Sim-to-real

ABSTRACT

High-performance controller tuning in industrial systems is still constrained by costly and time-consuming on-hardware experimentation. Although simulation models are widely available, they are often used only for offline verification and not as active information sources during optimization. This paper proposes the Adaptive Twin-in-the-Loop architecture, a framework that runs an ensemble of real-time digital twins in parallel with the physical plant to accelerate controller tuning in real applications. We develop a multi-source Bayesian Optimization method that jointly models real and virtual data through a multi-output Gaussian Process. A dedicated batch acquisition function then selects virtual evaluations that maximally reduce uncertainty in the most promising region of the real-system objective. Statistical validation on standard benchmarks and experimental testing on a brushless direct current motor speed controller show that the proposed framework accelerates convergence, reaching near-optimal closed-loop performance, on average, with only one-third of the on-hardware trials required by single-source Bayesian Optimization.

1. Introduction

The deployment of high-performance control systems in modern industrial applications remains a labor-intensive bottleneck. For complex electromechanical systems, such as Brushless Direct Current (BLDC) motors, achieving optimal tracking and stability is particularly challenging due to inherent non-linearities, unknown mechanical loads, and varying parameters like stator resistance or inertia (Kethiri et al., 2025; Mahmud et al., 2020). While initial controller parameters can be derived from classical models or tuned offline, these static configurations often degrade in performance when deployed on the real physical plant due to unmodeled load scenarios (Kethiri et al., 2025). Consequently, optimal tuning requires iterative, data-driven frameworks that interact directly with the physical plant to capture its exact operating conditions and unmodeled dynamics (Busetto et al., 2023). To safely and systematically execute this online interaction, the field has shifted toward sample-efficient zeroth-order frameworks (Larson et al., 2019) that optimize a performance metric through a limited number of physical trials, treating the closed-loop physical system as a black box.

Within this class of optimizers, Bayesian Optimization (BO) (Garnett, 2023), typically implemented with Gaussian Process (GP) surrogate models, is the most broadly utilized framework for controller tuning

(Neumann-Brosig et al., 2019; Nobar et al., 2024). Particularly for electromechanical systems and electric motor drives, BO has proven highly effective, achieving optimal closed-loop performance through data-driven evaluations (Khosravi et al., 2022). Its popularity has led to the development of multiple extensions, including constrained optimization (Gardner et al., 2014), safe exploration strategies designed to avoid unstable regions (Sui et al., 2015), risk-averse formulations that mitigate the influence of aleatoric uncertainty (Makarova et al., 2021), and, more recently, transfer-learning and simulation-informed variants that leverage prior tasks or auxiliary models to improve sample efficiency (Bai et al., 2023; Garces et al., 2024).

Despite the inherent sample efficiency of BO, many controller tuning implementations still rely primarily on the physical system as the main information source. This restriction is limiting because physical experimentation has a tangible cost in elapsed test time, hardware wear, and safety risk during exploration (Busetto et al., 2023). This motivates augmenting the optimization with virtual evaluations that are cheaper only relative to physical trials, since they shift effort from hardware experimentation to computation.

Over the last few years, accelerated by the momentum of Industry 4.0, virtual representations of physical systems have become widely deployed operational assets across the automotive, aerospace,

* Corresponding author.

E-mail addresses: giacomo.delcaro@polimi.it (G. Delcaro), simone2.gabrielli@polimi.it (S. Gabrielli), simone.formentin@polimi.it (S. Formentin).

¹ Present address: inspire AG and ETH Zürich, Zürich, Switzerland. This work was carried out while the author was with DEIB, Politecnico di Milano, Milano, Italy.

manufacturing, and energy sectors. However, many of these virtual representations operate strictly as offline Digital Models rather than true Digital Twins (DTs) (Grieves & Vickers, 2017; Kritzinger et al., 2018). By definition, a DT requires a continuous, bidirectional flow of information: data from the physical asset informs the virtual environment in real time, while optimized parameters, synthesized using both virtual and physical data, are fed back to control the physical plant (Sepasgozar, 2021). While offline models offer a risk-free testing environment, they cannot actively compensate for unmodeled physical phenomena or unknown mechanical loads. Therefore, integrating true DTs directly into the online optimization loop presents a powerful strategy.

A significant barrier to integrating virtual data into the tuning process is the discrepancy between simulated and physical responses, commonly referred to as the simulation-to-reality (sim2real) gap. This gap arises because DTs cannot fully capture real-world phenomena such as unmodeled dynamics, parameter drift, or noise. Consequently, controllers that exhibit high performance in virtual environments may underperform or become unsafe upon deployment on the physical plant. This paper introduces a multi-source BO (MSBO) framework designed to overcome this limitation by actively querying both the real system (RS) and an ensemble of randomized DTs running in parallel. The sim2real gap is addressed at the surrogate level by the Intrinsic Coregionalization Model (ICM), a symmetric multi-output Gaussian Process (MOGP) that represents the real system and all DTs jointly and learns statistical correlations among the corresponding objective functions during optimization. By explicitly modeling the sim2real gap, the optimizer continuously interacts with the real plant while leveraging concurrent virtual evaluations to reduce the model's uncertainty. This drastically decreases the number of real-world interactive evaluations required to safely tune the physical plant.

1.1. Related work

Related literature on data-efficient sim2real tuning and multi-source Bayesian Optimization spans several complementary directions.

Fröhlich et al. (2022) adopt a contextual BO viewpoint for Model Predictive Control (MPC) tuning in autonomous racing, showing how explicit context variables can be incorporated to adapt controller parameters quickly across operating conditions. However, this strategy requires a compact and measurable context representation. In our setting, the dominant context corresponds to latent plant parameters (e.g., load, friction, inertia mismatch) that are not directly measured.

Marco et al. (2017) present a BO framework that partitions queries between cheap, lower-fidelity simulations and time-consuming physical experiments using a multi-fidelity GP with a dual-kernel structure. Despite its demonstrated success, this strategy is limited to one virtual source and strictly sequential queries between digital and real systems.

Nobar et al. (2024) introduce Guided BO, a data-efficient algorithm that employs an event-triggered digital twin to perform virtual exploration when surrogate uncertainty is high. While effective in reducing physical experiments, the approach is limited to a single, sequential DT and lacks a multi-source surrogate capable of explicitly distinguishing between real and virtual data.

While previous approaches rely on a single DT evaluated sequentially with the real system, Allamaa et al. (2025) introduce a framework that executes an ensemble of DTs simultaneously. This architecture replaces the Bayesian surrogate with a combination of an Unscented Kalman Filter (UKF) and Simultaneous Perturbation Stochastic Approximation (SPSA) to perform derivative-free parameter updates. However, the absence of a Bayesian model precludes integration with risk-averse or safe BO variants and prevents the scheme from incorporating pre-simulated offline data.

Beyond the specific domain of control, prior BO methods that exploit auxiliary sources can be broadly divided into two groups. A first line of work addresses multi-fidelity modeling through hierarchical surrogate constructions, such as recursive co-kriging and multi-fidelity GP band-

dit optimization (Kandasamy et al., 2019; Le Gratiet & Garnier, 2014). These methods assume that the available sources can be organized as progressively more accurate approximations of the target objective. This assumption does not match our setting: the DTs are not naturally ordered into fidelity levels, since they all have the same structure and fidelity and differ mainly through randomized parameterizations introduced to span model uncertainty, rather than through a known hierarchy of accuracies. As a result, the problem considered here is not fidelity-level selection within a hierarchy, but coordinated information extraction from multiple correlated virtual sources.

A second line of work considers multiple tasks or information sources with cost-aware transfer, including multi-task BO and multi-information source optimization (Bai et al., 2023; Feurer et al., 2018; Leen et al., 2012; Poloczec et al., 2017; Swersky et al., 2013). These methods are relevant because they model cross-source correlations. However, their query policy is sequential: at each iteration, they select one source and one design point, while all other information sources remain unused. This limitation is critical in settings with multiple parallelizable sources. In this paper, we address it by adopting a parallel-query setting in which each iteration runs one experiment on the real system and concurrent experiments on multiple DTs.

1.2. Main contributions

In this article, we propose (C1) the Adaptive Twin-in-the-Loop framework (A-TiL), which uses multiple randomized DTs in parallel with the real system to achieve target performance with fewer on-hardware experiments, by shifting part of the exploration to virtual evaluations that do not consume additional plant trials. We develop (C2) an innovative multi-source BO algorithm that is able to jointly select samples from both the real system and the DTs, fully exploiting the parallel architecture of A-TiL. This is achieved by modeling the combined virtual and real data using the Intrinsic Coregionalization Model, a symmetric multi-output GP that learns cross-source correlations within the surrogate, supporting information transfer from DT evaluations to the real-system model, and by designing (C3) a novel multi-source batch acquisition function (AF) that uses virtual data to reduce model uncertainty in the region of potential improvement.

Statistical analysis on four benchmark functions shows that the strategy achieves significantly faster convergence than single-source BO methods, albeit with higher computational complexity. The method is then applied to a real control tuning problem for a speed controller of a BLDC motor, where the proposed framework (C4) achieves near-optimal performance on average in one-third of the on-hardware experiments required by single-source BO.

Structure of the article

This article is structured as follows. Section 2 introduces the Adaptive Twin-in-the-Loop architecture and optimization problem. Section 3 recalls Bayesian Optimization fundamentals, single-source Gaussian Processes and acquisition functions. Section 4 presents the multi-source GP surrogate, starting from the multi-output setting and then detailing the Intrinsic Coregionalization Model adopted in this work, while Section 5 describes the multi-source acquisition function designed for parallel query selection across the real system and the DTs. Section 6 reports controlled synthetic experiments to analyze convergence and computational behavior and Section 7 validates the same methodology on a BLDC motor testbed. Finally, Section 8 concludes the article and outlines future research directions.

2. Adaptive twin-in-the-loop architecture

The A-TiL scheme leverages an ensemble of P real-time executable DTs running in parallel. Each DT is defined by a different parameterization $\xi^{(p)}$ ($p = 1, \dots, P$) to capture different possible model uncertainties and parameter variations. This use of a parameterized ensemble is a

form of *Domain Randomization* (Peng et al., 2018; Tobin et al., 2017), a technique designed to bridge the inherent sim2real gap between any simulation environment and its physical counterpart. Rather than relying on a single, inevitably imperfect DT, the goal is to create a diverse set of models that collectively span the space of possible real-system behaviors. The chosen set of perturbed parameters should consist of those that significantly influence closed-loop stability and performance, particularly those subject to modeling errors. This ensures the digital twin ensemble provides a robust representation of the physical plant's behavior, effectively spanning the sim2real gap.

The objective is to tune a controller $C(\theta)$ parameterized by $\theta \in \Omega \subseteq \mathbb{R}^d$ by minimizing a closed-loop performance metric $J^{(RS)}(\theta)$ on the real system (e.g., tracking error, overshoot, phase margin). Formally, the goal is to find

$$\theta^* = \arg \min_{\theta \in \Omega} J^{(RS)}(\theta), \quad (1)$$

The DTs and RS operate in closed loop with a common reference signal $r(t)$, and evaluate different controller parameters: while the RS evaluates the cost $J^{(RS)}(\theta_i^{(RS)})$ for a specific controller parameter set $\theta_i^{(RS)}$, the DTs assess virtual costs $J^{(DT_p)}(\theta_i^{(DT_p)})$ for various parameter sets $\theta_i^{(DT_p)}$. These virtual evaluations act as lower-cost proxies relative to real-system experiments, as they can be generated without additional on-hardware trials.

Fig. 1 illustrates the A-TiL architecture: the lower area (green) shows the controller $C(\theta_i^{(RS)})$ closed on the real system. Using the reference $r(t)$, control action $u(t)$, and measurement $y(t)$, the real cost $J^{(RS)}(\theta_i^{(RS)})$ can be computed. The upper area (light blue) shows the parallel DTs; for each DT, the virtual cost function $J^{(DT_p)}(\theta_i^{(DT_p)})$ is computed. An adaptation law (orange, on the right) then uses real and virtual data as part of an active learning strategy to propose the next candidate controllers to test on the RS $\theta_{i+1}^{(RS)}$ and on the DTs $\theta_{i+1}^{(DT_p)}$. In this work, we propose a novel multi-source BO algorithm to implement this adaptation law, where learned correlations enable information transfer from virtual to real data. Finally, a batch multi-source acquisition function is designed to select the next candidate parameters to evaluate in parallel on the real system and on the DTs.

It is important to emphasize that the A-TiL framework employs true Digital Twins rather than offline Digital Models. Following the established taxonomy by Kritzinger et al. (2018), a DT requires a bidirectional flow of information with the physical asset. In our architecture, the DTs run in real time alongside the physical plant and share the same dynamic reference signals and inputs (tracking the real system). Simul-

taneously, the data generated by the DTs is continuously fed into the multi-source optimization algorithm to update the real plant's controller parameters.

3. Preliminaries

3.1. Bayesian optimization

Unlike sample-intensive zeroth-order methods such as Dividing RECTangles (DIRECT) (Jones, 2009), Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995), or genetic algorithms (Srinivas & Patnaik, 1994), Bayesian Optimization is an Efficient Global Optimizer (Jones et al., 1998) designed specifically for maximum sample efficiency.

After each sample, BO fits a probabilistic surrogate model to the objective function. This surrogate informs an acquisition function $a(\theta)$, a utility function that selects the next query point by balancing exploration (sampling uncertain regions) and exploitation (targeting predicted minima).

Single-Output Gaussian Processes. While various probabilistic models can serve as surrogates, GPs are favored for their analytically tractable posterior. Formally, a GP is a stochastic process defined as a distribution over functions, such that the distribution of the process at any finite set of input points is a multivariate Gaussian. It is fully specified by a mean function $m(\theta)$ and a covariance (kernel) function $k(\theta, \theta')$, and is denoted as

$$J^{(RS)}(\theta) \sim \mathcal{GP}(m(\theta), k(\theta, \cdot)). \quad (2)$$

In experimental settings, exact evaluations of the objective function are rarely accessible. Instead, we assume access to a dataset $\mathcal{D}^{(RS)} = (\Theta^{(RS)}, \bar{J}^{(RS)})$ of observed input-output pairs, where $\bar{J}^{(RS)}$ represents the cost evaluations at inputs $\Theta^{(RS)} = \{\theta_1^{(RS)}, \dots, \theta_n^{(RS)}\}$, corrupted by independent and identically distributed (i.i.d.) Gaussian noise with variance σ^2 . Conditioning the GP prior (2) on the observations $\mathcal{D}$ yields a closed-form posterior distribution:

$$J^{(RS)}(\theta) | \mathcal{D} \sim \mathcal{N}(\mu_{\text{post}}(\theta), \Sigma_{\text{post}}(\theta, \theta)), \quad (3a)$$

$$\mu_{\text{post}}(\theta) = m(\theta) + K(\theta, \Theta^{(RS)})C^{-1}(\bar{J}^{(RS)} - m(\Theta^{(RS)})), \quad (3b)$$

$$\Sigma_{\text{post}}(\theta, \theta') = K(\theta, \theta') - K(\theta, \Theta^{(RS)})C^{-1}K(\Theta^{(RS)}, \theta'), \quad (3c)$$

where $C = K(\Theta^{(RS)}, \Theta^{(RS)}) + \sigma^2 I$. The notation $K(A, B)$ refers to the covariance matrix computed between all pairs of points in sets A and B , such that $[K(A, B)]_{ij} = k(a_i, b_j)$.

To maintain computational efficiency, we adopt an empirical Bayes (or Type-II Maximum Likelihood) treatment for the GP hyperparameters ψ , which include the signal variance and the characteristic lengthscales of the kernel. This approach avoids computationally demanding numerical integration by instead maximizing the log-marginal likelihood of the observed data $p(\bar{J}^{(RS)} | \Theta, \psi)$ (Williams & Rasmussen, 2006).

Single-Source Acquisition Functions. The acquisition function $a(\theta)$ directs the search by quantifying the utility of sampling a specific candidate point. While various formulations exist, such as Expected Improvement (EI), Entropy Search (ES), and Knowledge Gradient (KG) (Garnett, 2023), this work adopts the Lower Confidence Bound (LCB):

$$a_{\text{LCB}}(\theta | \mathcal{D}) = -\text{lcb}(\theta) = -\left(\underbrace{\mu_{\text{post}}(\theta)}_{\text{exploitation}} - \beta \underbrace{\sigma_{\text{post}}(\theta)}_{\text{exploration}} \right), \quad (4)$$

where the outer minus sign transforms the minimization problem into a maximization one and the parameter $\beta > 0$ balances the exploration-exploitation trade-off. At every iteration, the next candidate is selected by solving $\theta_{i+1} = \arg \max_{\theta \in \Omega} a_{\text{LCB}}(\theta | \mathcal{D})$.

The algorithm for BO is summarized in Algorithm 1. Building on this single-source BO framework, we will develop a multi-source BO algorithm that can leverage both real and virtual data within the A-TiL architecture. This will be achieved by modeling the combined data using a multi-output Gaussian Process (Section 4) and designing a novel

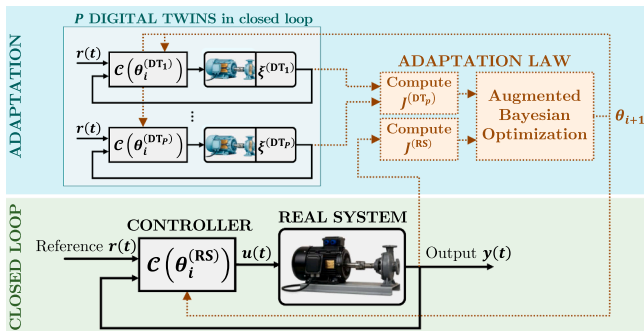


Fig. 1. Adaptive twin-in-the-loop architecture. The controller $C(\theta_i)$ is closed on the real system (lower area, green) and the real cost $J^{(RS)}(\theta_i^{(RS)})$ is computed. In parallel, P digital twins (upper area, light blue) run closed loop with the same reference signal $r(t)$ and evaluate virtual costs $J^{(DT_p)}(\theta_i^{(DT_p)})$. An adaptation law (orange) uses real and virtual data to propose the next candidate controllers to test on the RS $\theta_{i+1}^{(RS)}$ and on the DTs $\theta_{i+1}^{(DT_p)}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

acquisition function that selects samples from both sources in parallel (Section 5).

Algorithm 1: Classical Bayesian optimization.

Input: N_{BO} number of BO iterations, N_{seed} number of initial points, ψ GP hyperparameters.

// Initialization

1: Randomly sample N_{seed} seed points

2: Update dataset $D = \{(\theta_i, \bar{J}_i^{(\text{RS})})_{i=1}^{N_{\text{seed}}}\}$

// BO loop

3: **for** $i \leftarrow N_{\text{seed}} + 1$ **to** $N_{\text{BO}} + N_{\text{seed}}$ **do**
 // Optimize GP hypers and update posterior model

4: $\psi \leftarrow \arg \max_{\psi} \log p(\bar{J}^{(\text{RS})} | \Theta, \psi)$

5: $J^{(\text{RS})}(\theta) \leftarrow \text{Eq. (3a)} \text{ conditioned on } D$

 // Optimize acquisition function

6: $\theta_i \leftarrow \arg \max_{\theta \in \Omega} a(\theta | D)$

 // Evaluate objective function

7: Sample $\bar{J}_i^{(\text{RS})} = J^{(\text{RS})}(\theta_i) + \text{noise}$

8: Update dataset $D \leftarrow D \cup \{(\theta_i, \bar{J}_i^{(\text{RS})})\}$

9: **end**

10: **return** Best evaluated point $\theta^* = \arg \min_{\theta_i \in \Theta} \bar{J}_i^{(\text{RS})}$

4. Multi-source GP surrogate

4.1. Multi-output GP setting

In the proposed A-TiL setting, information is collected from multiple sources: the RS and an ensemble of DTs. To construct a GP surrogate from multiple sources, two main strategies are available: extending the input space with a source/context variable (as in Fröhlich et al., 2022) or modeling the sources as multiple outputs (Berkenkamp et al., 2016). In this work, we adopt the second strategy, and represent the objective as $\mathbf{J}(\theta) = [J^{(\text{RS})}(\theta), J^{(\text{DT}_1)}(\theta), \dots, J^{(\text{DT}_P)}(\theta)]^T \in \mathbb{R}^{P+1}$.

Multi-output GP models are commonly classified based on their dependency structure into symmetric and asymmetric models (Liu et al., 2018). Asymmetric models are suited to hierarchical outputs and are often used in multi-fidelity settings, where lower-fidelity outputs support faster learning of higher-fidelity outputs (Marco et al., 2017). Conversely, symmetric MOGPs are intended for outputs that are of equal importance, as they adopt bidirectional exchange of information. The two classes should not be considered mutually exclusive categories, as the most commonly adopted asymmetric model, the autoregressive MOGP, can be interpreted as a special case of the symmetric Semiparametric Latent Factor Model (as shown in §4.2.1 of Do & Zhang, 2025).

In our setting, the optimization target is the RS, but the available DTs are treated as auxiliary sources that are comparably informative with respect to one another. We therefore adopt a symmetric MOGP under the mild assumption that each DT has an objective function that shares the same smoothness properties as the RS objective function. This is consistent with multi-task BO settings, where data from related tasks is transferred to improve optimization on a target task using symmetric coregionalization models (Bonilla et al., 2007; Swersky et al., 2013). Section 6.2 will analyze the effect of uncorrelated DTs on the optimization performance.

In this work, we use the Intrinsic Coregionalization Model, which assumes a shared input kernel across outputs and source-specific coupling through a coregionalization matrix. Since our focus is optimization under scarce data rather than high-accuracy multi-output regression, this choice avoids the heavier hyperparameter burden of more expressive models such as the Linear Model of Coregionalization (Alvarez et al., 2012), which can be problematic in data-scarce scenarios. Future work will analyze more complex models and the problem of DTs-to-RS hierarchical transfer.

4.2. Intrinsic coregionalization model

Given the multi-output setting above, we now specify the concrete GP surrogate used throughout the paper. The ICM models the vector-valued objective function $\mathbf{J}(\theta)$ as a MOGP:

$$\mathbf{J}(\theta) = \begin{bmatrix} J^{(\text{RS})}(\theta) \\ J^{(\text{DT}_1)}(\theta) \\ \vdots \\ J^{(\text{DT}_P)}(\theta) \end{bmatrix} \sim \mathcal{GP}(\mathbf{m}(\theta), B * K(\theta, \cdot)), \quad (5)$$

where $\mathbf{m} \in \mathbb{R}^{P+1}$ is the GP prior mean, $B \in \mathbb{R}^{(P+1) \times (P+1)}$ the coregionalization matrix and $*$ the Khatri-Rao matrix product. In this study, the squared exponential kernel is adopted.

Let us assume that for each source $s = \{\text{RS}, \text{DT}_1, \dots, \text{DT}_P\}$, N_s (noisy) observations $\{(\theta_i^{(s)}, \bar{J}_i^{(s)})\}_{i=1}^{N_s}$ are available, where every output has its own independent (homoscedastic) noise:

$$\bar{J}_i^{(s)} = J^{(s)}(\theta_i^{(s)}) + \varepsilon_i^{(s)} \quad \text{with} \quad \varepsilon_i^{(s)} \sim \mathcal{N}(0, \sigma_s^2). \quad (6)$$

The samples and observations for each task can be collected in vectors

$$\Theta^{(s)} = \begin{bmatrix} \theta_1^{(s)} \\ \vdots \\ \theta_{N_s}^{(s)} \end{bmatrix} \in \mathbb{R}^{N_s \times d} \quad \text{and} \quad \bar{\mathbf{J}}^{(s)} = \begin{bmatrix} \bar{J}_1^{(s)} \\ \vdots \\ \bar{J}_{N_s}^{(s)} \end{bmatrix} \in \mathbb{R}^{N_s},$$

which can be stacked to form the complete dataset $D = (\Theta, \bar{\mathbf{J}})$, where

$$\Theta = \begin{bmatrix} \Theta^{(\text{RS})} \\ \vdots \\ \Theta^{(\text{DT}_P)} \end{bmatrix} \in \mathbb{R}^{N \times d} \quad \text{and} \quad \bar{\mathbf{J}} = \begin{bmatrix} \bar{\mathbf{J}}^{(\text{RS})} \\ \vdots \\ \bar{\mathbf{J}}^{(\text{DT}_P)} \end{bmatrix} \in \mathbb{R}^N.$$

Once conditioned with the data D , the GP model posterior becomes:

$$\mathbf{J}(\theta) | \theta, D \sim \mathcal{N}(\mu_{\text{post}}(\theta), \Sigma_{\text{post}}(\theta, \theta)), \quad (7a)$$

$$\mu_{\text{post}}(\theta) = \mathbf{m}(\theta) + (B * \underline{K}_{\Theta\Theta}) C_{MO}^{-1} (\bar{\mathbf{J}} - \mathbf{m}(\Theta)), \quad (7b)$$

$$\Sigma_{\text{post}}(\theta, \theta') = (B * \underline{K}_{\theta\theta'}) - (B * \underline{K}_{\theta\Theta}) C_{MO}^{-1} (B * \underline{K}_{\Theta\theta'})^T, \quad (7c)$$

where $C_{MO} = B * \underline{K}_{\Theta\Theta} + \Sigma_n$, $\Sigma_n \in \mathbb{R}^{N \times N}$ is the noise covariance matrix: $\Sigma_n = \text{diag}(\sigma_1^2 I_{N_1}, \dots, \sigma_{P+1}^2 I_{N_{P+1}}) \in \mathbb{R}^{N \times N}$. The covariance matrices $\underline{K}_{\Theta\Theta}$, $\underline{K}_{\theta\Theta}$ and $\underline{K}_{\theta\theta'}$ are block matrices with $(P+1) \times (P+1)$ blocks to match the dimensions of B . They are computed as follows:

$$\underline{K}_{\theta\theta'} = \mathbf{1}_{(P+1) \times (P+1)} \otimes K(\theta, \theta')$$

$$\underline{K}_{\theta\Theta} = \mathbf{1}_{(P+1) \times 1} \otimes [K(\theta, \Theta^{(\text{RS})}), K(\theta, \Theta^{(\text{DT}_1)}), \dots, K(\theta, \Theta^{(\text{DT}_P)})]$$

$$\underline{K}_{\Theta\Theta} = \begin{bmatrix} K(\Theta^{(\text{RS})}, \Theta^{(\text{RS})}) & K(\Theta^{(\text{RS})}, \Theta^{(\text{DT}_1)}) & \dots & K(\Theta^{(\text{RS})}, \Theta^{(\text{DT}_P)}) \\ K(\Theta^{(\text{DT}_1)}, \Theta^{(\text{RS})}) & K(\Theta^{(\text{DT}_1)}, \Theta^{(\text{DT}_1)}) & \dots & K(\Theta^{(\text{DT}_1)}, \Theta^{(\text{DT}_P)}) \\ \vdots & \vdots & \ddots & \vdots \\ K(\Theta^{(\text{DT}_P)}, \Theta^{(\text{RS})}) & K(\Theta^{(\text{DT}_P)}, \Theta^{(\text{DT}_1)}) & \dots & K(\Theta^{(\text{DT}_P)}, \Theta^{(\text{DT}_P)}) \end{bmatrix}$$

where $K(\cdot, \cdot)$ is the covariance matrix computed using the kernel function $k(\cdot, \cdot)$ as in Section 3.1, $\otimes$ is the Kronecker product and $\mathbf{1}_{a \times b}$ is an $a \times b$ matrix of ones.

4.3. Coregionalization matrix parameterization

To further minimize the hyperparameter count and ensure numerical stability, we parameterize B using the continuous relaxation method (Saves et al., 2023), where tasks are represented by one-hot encodings $e_s = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^{P+1}$ (the s -th entry is 1 and all the others are 0). Consequently, the correlation between two sources s and s' can be expressed as $b_{ss'} = \text{cov}(e_s, e_{s'})$. Since e_s and $e_{s'}$ are vectors, their covariance can be modeled using a canonical kernel function. Using the Automatic Relevance Determination (ARD) squared exponential kernel, the entries of B become:

$$b_{ss'} = \begin{cases} 1 & \text{if } s = s' \\ \exp(-1/2 \cdot (\ell_s^{-2} + \ell_{s'}^{-2})) & \text{if } s \neq s' \end{cases}, \quad (8)$$

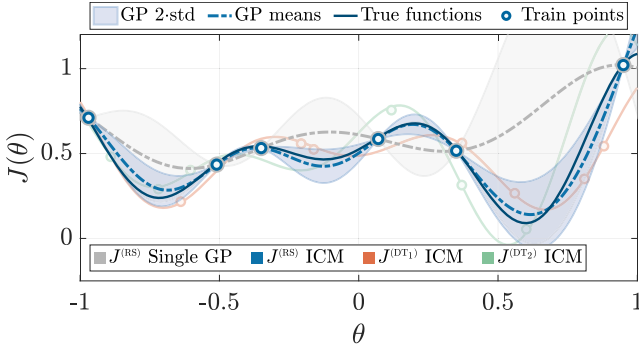


Fig. 2. Illustrative ICM GP example with correlated but distinct sources. The plot displays the posterior mean (dashed lines) and $\pm 2\sigma$ confidence intervals (shaded regions) for the target $J^{(RS)}$. The single-output GP (grey) is only conditioned on $J^{(RS)}$ samples while the ICM MOGP (blue) also relies on data from $J^{(DT_1)}$ (orange points) and $J^{(DT_2)}$ (green points). The ICM achieves significantly lower uncertainty and a mean estimate that more accurately tracks the true function (solid line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where ℓ_s and $\ell_{s'}$ are the lengthscales associated with sources s and s' , and they are learned during marginal likelihood optimization. This parameterization allows one to model inter-source correlations with only $P + 1$ hyperparameters (one lengthscale per source), significantly reducing the number of hyperparameters compared to a full B matrix parameterization, which would require $(P + 1)(P + 2)/2$ parameters (Bonilla et al., 2007).

Overall, the ICM MOGP requires optimizing the classic GP hyperparameters (signal variance, lengthscales, noise variances) plus $P + 1$ lengthscales for the coregionalization matrix. Analogously to Algorithm 1, hyperparameters are learned by maximizing the log marginal likelihood of the observed data D : $p(\mathbf{J} | \Theta, \psi)$, where ψ is the set of all hyperparameters.

4.4. ICM MOGP illustrative example

Fig. 2 illustrates a simple 1D example of ICM MOGP compared to single-output GP. Three distinct information sources are considered: $J^{(RS)}$ (blue for ICM, grey for single-output GP), $J^{(DT_1)}$ (orange) and $J^{(DT_2)}$ (green). Each sample is color-matched to its source. Only posteriors for the target source $J^{(RS)}$ are shown. Since the ICM leverages data from all sources, the resulting posterior has lower uncertainty (shaded area) and a mean estimate (dashed line) much closer to the true function (solid line), when compared to the single-output GP. The higher the inter-source correlation, the more pronounced the information transfer and the resulting uncertainty reduction.

Once the multi-source surrogate model is defined, the next Section is devoted to the design of an acquisition function that selects candidates from all the information sources.

5. Multi-source acquisition functions

5.1. Batch acquisition background

In our multi-source setting, each BO iteration can exploit one RS query and multiple DT queries in parallel. This requires a batch decision rule across candidate evaluations. Classical acquisition functions (LCB, EI, ES, KG) select one point per iteration, whereas batch acquisition functions generalize the selection to q points in parallel.

Early batch methods were mainly greedy: GP-BUCB (Batch Upper Confidence Bound, Desautels et al., 2014) iteratively selected points with hallucinated outcomes from the posterior mean, while GP-UCB-PE (Contal et al., 2013) combined one UCB point with subsequent Pure Exploration points in promising regions. These strategies are computationally

efficient, but their sequential construction can be suboptimal because early selections constrain the rest of the batch. Joint batch optimizers, such as batch Entropy Search (Shah & Ghahramani, 2015) and q -EI (Wang et al., 2019), mitigate this issue at the cost of higher-dimensional and often Monte Carlo-intensive optimization.

In the present problem, the objective is not only to select multiple points, but also to allocate them across heterogeneous sources, an area largely unexplored in current literature. Existing multi-task BO formulations (e.g., Marco et al., 2017; Swersky et al., 2013) remain fundamentally sequential, as they select one point from one source at a time. To our knowledge, a parallel multi-source batch AF that jointly selects multiple candidates from multiple sources has not been previously reported.

5.2. Proposed multi-source acquisition functions

Using the multi-source ICM posterior defined in Section 4.2, we now define the decision mechanism that maps uncertainty into candidate points for the RS and DTs. In the A-Til framework, the RS is the target source and the DTs are auxiliary sources. Accordingly, we split the acquisition mechanism: the RS point is selected with a single-point AF on the RS posterior, while DT points are jointly selected to improve the RS model. The resulting multi-source BO procedure is summarized in Algorithm 2, while the associated AF optimization routine is detailed in Algorithm 3.

RS AF. As acquisition function for the RS, we employ the LCB:

$$\theta_{i+1}^{(RS)} = \arg \max_{\theta \in \Omega} [-\text{lcb}_i(\theta)], \quad (9)$$

$$\text{lcb}_i(\theta) = \mu_{\text{post}}^{(RS)}(\theta) - \beta \sqrt{\Sigma_{\text{post}}^{(RS)}(\theta)}, \quad (10)$$

where $\mu_{\text{post}}^{(RS)}(\theta) = \mathbf{e}_1^\top \mu_{\text{post}}(\theta)$, $\Sigma_{\text{post}}^{(RS)}(\theta) = \mathbf{e}_1^\top \Sigma_{\text{post}}(\theta) \mathbf{e}_1$ and $\mathbf{e}_1 = [1, 0, \dots, 0]^\top \in \mathbb{R}^{P+1}$ selects the RS source.

DT AF. A naive strategy for the DTs would be to select the same point for RS and DTs. However, this presents two main issues. First, it can cause autokrigability problems in the ICM (see §2.3 in Bonilla et al., 2007), where the inter-source correlations cancel out. Second, sampling the same point on different sources is inefficient, as evaluating a point on a related task can never reveal more information than evaluating the same point on the task of interest (RS).

In this work, we propose an approach similar to the Pure Exploration of GP-UCB-PE by Contal et al. (2013), expanded to a multi-source setting and reformulated as a batch optimization rather than a greedy one. The DT samples are selected to explore the uncertain region R_i of the model that potentially contains the optimum. This region is defined as the set of points where the RS lower confidence bound is below the minimum of the upper confidence bound:

$$R_i = \left\{ \theta \in \Omega \mid \text{lcb}_i(\theta) \leq \min_{\theta' \in \Omega} \text{ucb}_i(\theta') \right\}. \quad (11a)$$

where ucb_i is defined as in Eq. (10) but with a plus sign before the variance term. This region R_i encompasses all points that could potentially yield better objective values than the best upper confidence bound found so far, thus focusing exploration on the most promising areas of the search space. By construction, the next RS sample $\theta_{i+1}^{(RS)}$ always lies within this relevant region: $\theta_{i+1}^{(RS)} \in R_i$, as $\min_{\theta \in \Omega} \text{lcb}_i(\theta) \leq \min_{\theta' \in \Omega} \text{ucb}_i(\theta')$.

The DT incumbents are then selected to maximally reduce the RS posterior variance over the relevant region R_i :

$$\theta_{i+1}^{(DT_1)}, \dots, \theta_{i+1}^{(DT_P)} = \arg \max_{\theta_1, \dots, \theta_P \in R_i} d^{(DT)}(\theta_1, \dots, \theta_P \mid D^*),$$

being

$$d^{(DT)}(\theta_1, \dots, \theta_P \mid D^*) = \int_{R_i} \underbrace{\left(\Sigma_{\text{post}}^{(RS)}(\theta \mid D^*) - \Sigma_{\text{post}}^{(RS)}(\theta \mid D^* \cup \{\theta_p^{(DT_p)}\}_{p=1}^P) \right)}_{\text{variance before sampling DTs}} d\theta, \quad (12)$$

Algorithm 2: Multi-source BO.

Input: $S = \{RS, DT_1, \dots, DT_P\}$ sources, $N_{\max}$ BO iterations, $N_{\text{seed}}^{(s)}$ ($s \in S$) initial points per source, ψ ICM GP hyperparameters.

// Initialization (parallelized over sources)

1: **parfor** $s \in S$ **do**

2: Randomly sample $N_{\text{seed}}^{(s)}$ seed points $\theta_i^{(s)}$

3: Update dataset $D^{(s)} \leftarrow \{(\theta_i^{(s)}, \bar{J}_i^{(s)})_{i=1}^{N_{\text{seed}}^{(s)}}\}$

4: **end**

5: $D \leftarrow \bigcup_{s \in S} D^{(s)}$

// BO loop

6: **for** $i \leftarrow N_{\text{seed}}^{(RS)} + 1$ **to** $N_{\max} + N_{\text{seed}}^{(RS)}$ **do**

 // Optimize ICM hypers and update posterior

$\psi \leftarrow \arg \max_{\psi} \log p(\mathbf{J} | \Theta, \psi)$

$\mathbf{J}(\theta) \leftarrow \text{Eq. (7a)} \text{ conditioned on } D$

 // Optimize multi-source acquisition function

$(\theta_i^{(RS)}, \theta_i^{(DT_1)}, \dots, \theta_i^{(DT_P)}) \leftarrow \text{OPTMSACQFCN}(\mathbf{J})$

 // Evaluate objective functions (parallelized)

parfor $s \in S$ **do**

 11: Sample $\bar{J}_i^{(s)} = J^{(s)}(\theta_i^{(s)}) + \text{noise}$

 12: Update dataset $D^{(s)} \leftarrow D^{(s)} \cup \{(\theta_i^{(s)}, \bar{J}_i^{(s)})\}$

 13: **end**

 14: $D \leftarrow \bigcup_{s \in S} D^{(s)}$

15: **end**

16: **return** Best evaluated point for the RS $\theta^* = \arg \min_{\theta \in \Theta^{(RS)}} \bar{J}_i^{(RS)}$

where $\Sigma_{\text{post}}^{(RS)}(\theta | D^*)$ is the posterior variance of the RS evaluated at point θ , computed using Eq. (7c), while $\Sigma_{\text{post}}^{(RS)}(\theta | D^* \cup \{\theta_p^{(DT_P)}\}_{p=1}^P)$ is the posterior after adding the new DT samples. $D^* = D \cup \{\theta_{i+1}^{(RS)}\}$ represents the dataset augmented with the new RS sample optimized in Eq. (9). We use this updated dataset to account for the impending uncertainty reduction at that point. Neglecting this update could lead the acquisition function to select DT points close to $\theta_{i+1}^{(RS)}$ (the RS sample), yielding redundant information and inefficiently exploring the space, as shown in A.2.

Overall, the chosen acquisition function aims to select DT points that maximally reduce the uncertainty of the RS model in the most promising region R_i , thus improving the model where it matters most.

Eq. (12) also explains why DT samples do not collapse into the same region. The P DT points are optimized *jointly* as one batch, so two nearby candidates yield strongly redundant covariance updates and a smaller integrated RS-variance reduction over R_i . As a consequence, the optimizer naturally favors spatially separated and information-complementary DT queries.

The variance reduction in Eq. (12) can be computed efficiently using the following identity:

$$\begin{aligned} \Sigma_{\text{post}}(\theta | D^*) - \Sigma_{\text{post}}(\theta | D^* \cup \{\theta_p^{(DT_P)}\}_{p=1}^P) &= \\ = \Sigma_{\text{post}}(\theta, \Theta^*) (\Sigma_{\text{post}}(\Theta^*, \Theta^*) + \Sigma_n)^{-1} \Sigma_{\text{post}}(\theta, \Theta^*)^T, \end{aligned} \quad (13)$$

where $\Theta^* = [\Theta^T \ \theta_{i+1}^{(RS)}]^T$ is the augmented set of training points including the new RS sample. The matrix inversion in (13) can be precomputed before optimizing the acquisition function, thus reducing the computational burden.

Unlike outcome-based batch AFs, Eq. (12) depends only on GP variance and therefore does not require marginalization over pending outcomes. This is a key computational advantage over EI/ES/KG-style multi-source batch extensions, which evaluate expected utility by integrating over unobserved outcomes. Our method still requires integration over the input region R_i via Monte Carlo (MC) sampling, as discussed in A.1, but this step is typically lighter than full outcome marginalization. Compared with existing approaches, this rule is novel in combining *parallel* source allocation and batch-point selection in a single decision step:

Algorithm 3: Optimize multi-source acq. Fcn.

1: **Function** OPTMSACQFCN($\mathbf{J}$)

 // Optimize RS Candidate (Eq. 9)

2: $\theta^{(RS)} \leftarrow \arg \max_{\theta \in \Omega} [-\text{lcb}_i(\theta)]$

3: $D^* \leftarrow D \cup \{\theta^{(RS)}\}$

 // Define Relevant Region R_i (Eq. 11a)

4: $\text{ucb} \leftarrow \min_{\theta \in \Omega} \text{ucb}_i(\theta)$

5: Generate a set of N_{samples} candidate points $\Theta_{\text{samples}} \subset \Omega$ using LHS

6: $R_i \leftarrow \{\theta \in \Theta_{\text{samples}} | \text{lcb}_i(\theta) \leq \text{ucb}\} \cup \{\theta^{(RS)}\}$

7: **if** $|R_i| < N_{\text{min samples}}$ **then**

 // Build boxes $\mathcal{H}$ around points in R_i and sample

8: $\mathcal{H} = \bigcup_{\theta_j \in R_i} H_j$, where $H_j = [\mathbf{l}(\theta_j), \mathbf{u}(\theta_j)]$ and each dimension k of $\mathbf{l}, \mathbf{u}$ is obtained as

$u_k(l_k) \leftarrow \theta_j \pm \sup_{a \geq 0} \{\text{lcb}(\theta_j \pm a e_k) \leq \text{ucb}\}$

9: Sample $N_{\text{min samples}} - |R_i|$ points from $\mathcal{H}$ using LHS and add them to R_i

10: **end**

 // Generate candidate batches and evaluate DT AF

11: Draw N_{raw} batches $\Phi_1, \dots, \Phi_{N_{\text{raw}}}$ of P distinct points from R_i each

12: Evaluate $v_j \leftarrow a^{(DT)}(\Phi_j | D^*)$ for $j = 1, \dots, N_{\text{raw}}$

 // Use best batches as starting points for local optimiz.

13: Identify indices $\mathcal{J}$ of the N_{restart} largest v_j

14: $(\theta^{(DT_1)}, \dots, \theta^{(DT_P)}) \leftarrow \arg \text{local max } a^{(DT)}(\Phi | D^*)$ initialized at Φ_j , for each $j \in \mathcal{J}$

15: **return** $(\theta^{(RS)}, \theta^{(DT_1)}, \dots, \theta^{(DT_P)})$

one RS point is selected by LCB, while P DT points are jointly selected to maximize RS variance reduction over R_i .

AF Optimization Procedure. To optimize the AF, we use multiple random restarts and local optimization, as performed in most BO frameworks (as Balandat et al., 2020). The procedure is reported in Algorithm 3 and summarized below:

- Sample N_{samples} quasi-random (Latin Hypercube Sampling, LHS) points within the optimization variable space Ω (line 5 of Alg. 3).
- Identify the subset of these points belonging to the relevant region R_i of Eq. (11a) (line 6 of Alg. 3).
- Randomly select N_{raw} combinations of P points from the set in b, and evaluate each combination using the DT AF of Eq. (12) (lines 11-12 of Alg. 3).
- Use the N_{restart} best combinations as starting points for local constrained optimization (Limited-memory Broyden-Fletcher-Goldfarb-Shanno, L-BFGS) of the AF (lines 13-14 of Alg. 3).

The most computationally intensive software-side steps are c and d, as they require multiple evaluations of the DT AF.

If the search space is large relative to the relevant region R_i , the initial quasi-random sampling may yield an insufficient number of points within R_i , rendering step c infeasible. In such cases, we employ a heuristic that locally approximates R_i using hyper-rectangles centered on valid samples, from which additional candidates are drawn (lines 7-10 of Alg. 3). This also motivates the LCB choice: as established above, the RS query belongs to R_i by construction, so at least one valid center for the hyper-rectangle heuristic is always available. If an alternative AF such as EI or KG were utilized for the RS, finding a feasible initial point satisfying a similarly bounded probable-improvement region would require solving an additional, potentially complex, constraint satisfaction problem.

The complete procedure for our proposed Multi-Source BO algorithm is summarized in Algorithms 2 and 3.

5.3. AF illustrative examples

Perfectly correlated sources. Fig. 3 illustrates the acquisition function for a case of perfect correlation ($J^{(RS)} = J^{(DT_1)} = J^{(DT_2)}$). The top subplot (Fig. 3a) shows the GP posterior for the RS, with the relevant region R_i highlighted in grey. The red star indicates the next RS sample (LCB minimum), while vertical lines denote the next DT samples, targeting high-uncertainty areas in R_i . Training points are shown for RS (blue), DT1 (orange), and DT2 (green). The bottom subplots (Fig. 3b) present a contour plot of the DT AF (Eq. 12) for the two DTs. R_i is visualized as transparent cutouts within the shaded overlay that covers the rest of the parameter space. A few key observations can be made:

- The AF shows horizontal and vertical valleys close to previously sampled points (represented as circles on the top and left margins), as sampling close to already acquired points offers minimal new information.
- The AF is low along the diagonal, penalizing DT points close to each other, which would provide redundant information.
- The AF is highest where both DTs are far from existing samples, promoting exploration. The red cross indicates the selected DT samples (maximum of the DT AF in R_i).

The top and left marginal plots of Fig. 3b illustrate the DTs-to-RS covariance: the light blue line shows the covariance (7c) conditioned only on D , while the dark blue line shows the covariance conditioned on the augmented dataset D^* . These plots reveal that including the next RS sample $\theta_{i+1}^{(RS)}$ in the dataset significantly alters the DTs-to-RS covariance, effectively discouraging DT sampling in that vicinity. Overall, the DT AF effectively identifies points that maximize uncertainty reduction in the RS within R_i , while avoiding redundant sampling.

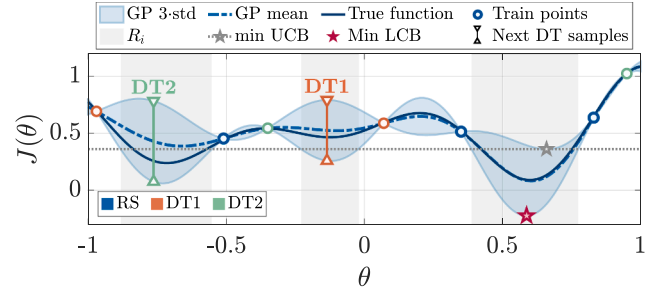
Correlated but distinct sources. Fig. 4 illustrates the acquisition function for the correlated but distinct sources case of Fig. 2. Here, vertical and horizontal valleys are present only close to points corresponding to each specific DT source, as the sources provide different information to the RS. The AF surface is asymmetric due to unequal source correlations, and the bisector is no longer a low-AF region, as different sources contribute different information to the RS.

6. Results with synthetic functions

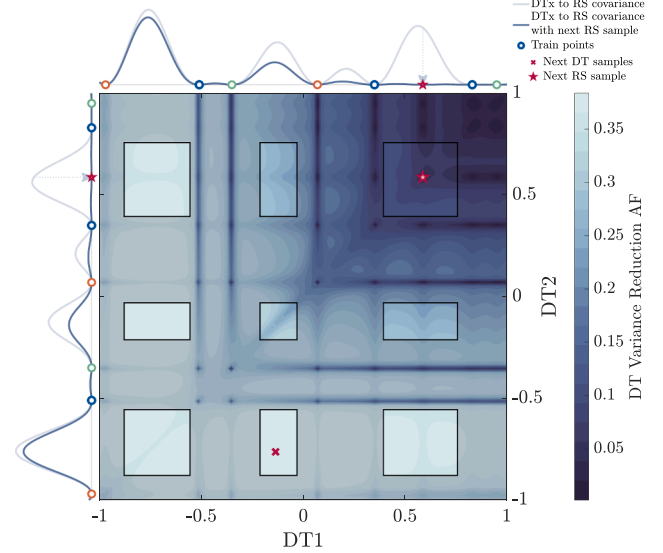
To connect the methodological development of Sections 4.2 and 5 with empirical evidence, we first evaluate the proposed method on controlled synthetic functions before moving to the physical testbed. We apply the MSBO algorithm in three cases: $P = 0$ (standard BO on the RS only), $P = 3$, and $P = 6$ DTs.

The synthetic tests are designed to isolate the key behaviors claimed by the method: convergence speed, uncertainty reduction from DT information, and algorithmic computational overhead. Within this section, we first report baseline convergence and runtime behavior, then provide dedicated sensitivity analyses and an ablation study on weakly informative sources. We consider four synthetic functions with different dimensionality and smoothness, commonly used as benchmarks in BO literature: (i) Ackley (2D), exhibiting thousands of local minima, (ii) Hartmann (3D), smoother than Ackley but with one additional dimension, (iii) Hartmann (6D), generalizing the 3D case, and (iv) Griewank (12D), to test high-dimensional optimization. For each benchmark function, the real system $J^{(RS)}$ is taken to be the canonical function definition. The DTs $J^{(DT_p)}$ are instead generated by multiplying the canonical parameters by a (scaled) random normal variable. Table 1 summarizes the definitions, parameterizations and optimization domains of each function, while Table 2 reports the MSBO hyperparameters adopted.

For each optimization problem, 30 independent optimizations are run with different random seeds for statistical significance. Fig. 5 reports the convergence plots of the best evaluation so far (best-so-far, bsf) after each iteration. The MSBO algorithm (blue and orange lines)



(a) GP posterior for the RS. The grey background highlights the region R_i . The red star indicates the next RS sample (LCB minimum), while vertical lines denote the next DT samples, targeting high-uncertainty areas in R_i . Training points are shown for RS (blue), DT1 (orange), and DT2 (green).

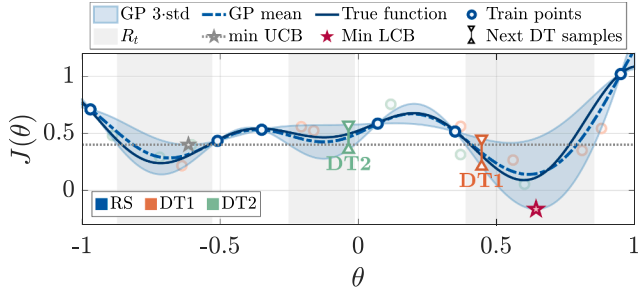


(b) Variance reduction acquisition function contour plot for DT1 vs. DT2. R_i is visualized as transparent cutouts within the shaded overlay that covers the rest of the parameter space. The red cross indicates the selected DT samples (maximum of the DT AF in R_i), while the red star marks the next RS sample. The top and left marginal plots illustrate the DTs-to-RS covariance: the light blue line shows the current covariance, while the dark blue line shows the covariance updated with the next RS sample. Sampled points are projected as circles color-coded by source. The AF shows low values (darker regions) along the bisector and in correspondence with previous samples, thereby discouraging data redundancy, ensuring the next DT samples are distinct from prior samples and from each other. The contour plot is symmetric with respect to the bisector due to perfect source correlation.

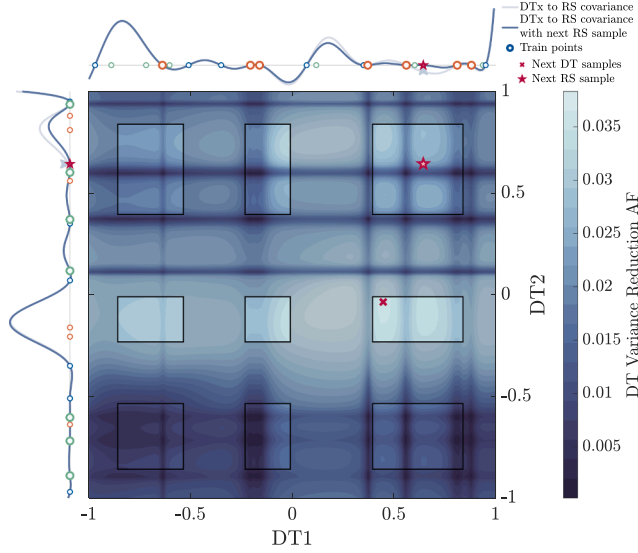
Fig. 3. DT acquisition function for a perfect correlation case.

outperforms standard BO (yellow lines), achieving both faster convergence and reduced optimization variability. The higher the complexity of the function (in terms of dimensionality and number of local minima), the greater the performance gain achieved by using multiple DTs. This is particularly evident in the Hartmann (6D) and Griewank (12D) functions, where the explorative information provided by the DTs significantly accelerates convergence. Going from $P = 3$ to $P = 6$ DTs generally provides further performance improvements, although the gain is far less pronounced than that of going from $P = 0$ to $P = 3$. This suggests diminishing returns when adding DTs, likely due to the simplicity of the ICM, which is limited in capturing complex inter-source relationships.

Computational times. Fig. 6 breaks down the mean iteration time for each function and DT configuration. These times refer to optimization-side computation (software runtime) in the synthetic benchmarks. The computation time is divided into three components: (i) objective evaluations (orange), (ii) GP fitting (hyperparameter optimization and pos-



(a) GP posterior for the RS.



(b) Variance reduction acquisition function. Due to unequal source correlations, the surface is asymmetric and the bisector is no longer a low-AF region, as the sources provide different information to the RS. The AF shows low values only in correspondence with the samples on the specific DT source.

Fig. 4. Acquisition function for the correlated but distinct sources case of Fig. 2.

terior update) (green), and (iii) AF optimization (Algorithm 3). Since the objectives are synthetic functions, their computation time is negligible, while AF optimization dominates the total computation time. As dimensionality and the number of DTs increase, the computation time lengthens. In terms of convergence improvement per unit time, 3 DTs offer a better trade-off than 6 DTs, as the marginal performance gain is limited while the computational cost approximately doubles.

Fig. 7 further details the computational time over the iterations for the Griewank function with 3 DTs, highlighting how AF optimization and GP fitting times grow with iterations as the dataset expands.⁴

6.1. Acquisition function sensitivity analyses

Offline data availability. We investigate the sensitivity of MSBO to the number of offline DT seed samples, a practically relevant setting since DTs can be pre-simulated before on-hardware optimization. Three configurations are tested on Ackley with 3 DTs: $N_{\text{seed}}^{(\text{DT})} \in \{1, 10, 100\}$ and the results are shown in Fig. 8. In particular, the bottom left panel of Fig. 8 reports the learned DTs-to-RS correlations: the solid line represents the mean DTs-to-RS correlation, the dotted curves denote the corresponding highest and lowest correlations along the iterations, averaged across 30 runs. The $N_{\text{seed}}^{(\text{DT})} = 1$ and 10 profiles are close, with the 10-seed case showing consistently higher DTs-to-RS correlations, indicating slightly more effective transfer; this is consistent with the small convergence advantage of 10 seeds (top left panel of Fig. 8). The $N_{\text{seed}}^{(\text{DT})} = 100$ case is more distinctive: correlations remain near zero for about the first 10 it-

Table 1
DT parameterization for each benchmark.

Function	Equation
	Nominal parameters
	DT perturbations $\xi_i^{(p)} = \xi_i(1 + \epsilon_i^{(p)})$
Ackley (2D)	$J(\theta) = a \left(1 - \exp \left(-b \sqrt{\frac{1}{d} \sum_{i=1}^d \theta_i^2} \right) \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(c \theta_i) \right) + e^1$ $a=20, b=0.2, c=2\pi$; domain $[-20, 20]^2$ $\xi = [a, b, c]^T$; $\epsilon_{\{a,b\}}^{(k)} \sim \mathcal{N}(0, 0.10^2)$, $\epsilon_c^{(k)} \sim \mathcal{N}(0, 0.05^2)$.
Hartmann (3D)	$J(\theta) = -\sum_{i=1}^4 c_i \exp \left(-\sum_{j=1}^3 A_{ij} (\theta_j - P_{ij})^2 \right)$ $c \in \mathbb{R}^4$, $A, P \in \mathbb{R}^{4 \times 3}$ as in (Jamil & Yang, 2013); domain $[0, 1]^3$ $\xi = [c, A_{ij}, P_{ij}]$; $\epsilon^{(k)} \sim \mathcal{N}(0, 0.10^2)$.
Hartmann (6D)	$J(\theta) = -\frac{m_0}{m_1} - \frac{1}{m_1} \sum_{i=1}^4 c_i \exp \left(-\sum_{j=1}^6 A_{ij} (\theta_j - P_{ij})^2 \right)$ $c \in \mathbb{R}^4$, $A, P \in \mathbb{R}^{4 \times 6}$ as in (Jamil & Yang, 2013); $m_0=2.58, m_1=1.94$; domain $[0, 1]^6$ $\xi = [c, A_{ij}, P_{ij}]$; $\epsilon^{(k)} \sim \mathcal{N}(0, 0.10^2)$.
Griewank (12D)	$J(\theta) = \sum_{i=1}^d \frac{\theta_i^2}{d_i} - \prod_{i=1}^d \cos \left(\frac{\theta_i}{\sqrt{i}} \right) + 1$ $d_1=4000, s_p=1, s_c=1$; domain $[-5, 5]^{12}$ $\xi = [d_1, s_p, s_c]^T$; $\epsilon^{(k)} \sim \mathcal{N}(0, 0.10^2)$.

Table 2
Experimental parameters for the benchmark functions.

Parameter	Ackley (2D)	Hart3 (3D)	Hart6 (6D)	Griewank (12D)
N_{samples}	1000	2000	10,000	20,000
N_{raw}	250	500	1000	2000
Common Parameters: (unless specified)	$N_{\text{seed}}^{(\text{RS})} = N_{\text{seed}}^{(\text{DT})} = 1$, $N_{\text{restart}} = 2$, $\beta = 3$, $N_{\text{min samples}} = 1.5 \cdot N_{\text{raw}}$			

erations, indicating that the optimizer initially struggles to align DT and RS information. Around iteration 10, all DTs-to-RS correlations increase sharply, and this transition coincides with the jump observed in the convergence plot, showing that once sufficient RS information is available the model can exploit cross-source correlations and ultimately outperform the other configurations. These results suggest that while offline data is beneficial for warm-starting, an excessively large initial dataset can temporarily suffer from “correlation lag” and slow down the convergence, despite providing superior long-term accuracy. Naturally, these performance gains are accompanied by an increase in total algorithmic runtime due to the larger dataset, as shown on the right panel of Fig. 8. **DT ensemble size (batch-size sensitivity).** In our formulation, the number of DT evaluations per iteration coincides with the DT batch size P . We now analyze this sensitivity using $P \in \{3, 10, 20\}$, with results shown in Fig. 9. The convergence panel in the top left of Fig. 9 shows that $P = 10$ yields the best performance, while $P = 3$ is slightly slower and $P = 20$ degrades. The correlation panel in the bottom left of Fig. 9 clarifies this trend: $P = 10$ achieves the highest DTs-to-RS correlations, $P = 3$ exhibits slightly lower initial correlations, and $P = 20$ remains close to zero for most iterations. This indicates that, with limited data, the ICM struggles to reliably correlate a large number of sources, reducing transfer effectiveness and worsening convergence. The software runtime (right panel in Fig. 9) also increases significantly with P , primarily due to AF optimization and larger multi-source GP updates. Overall, the trade-off is non-monotonic: increasing P can improve convergence only while correlations remain reliably learnable, after which model quality degrades and computational cost grows. These sensitivity results motivate the ab-

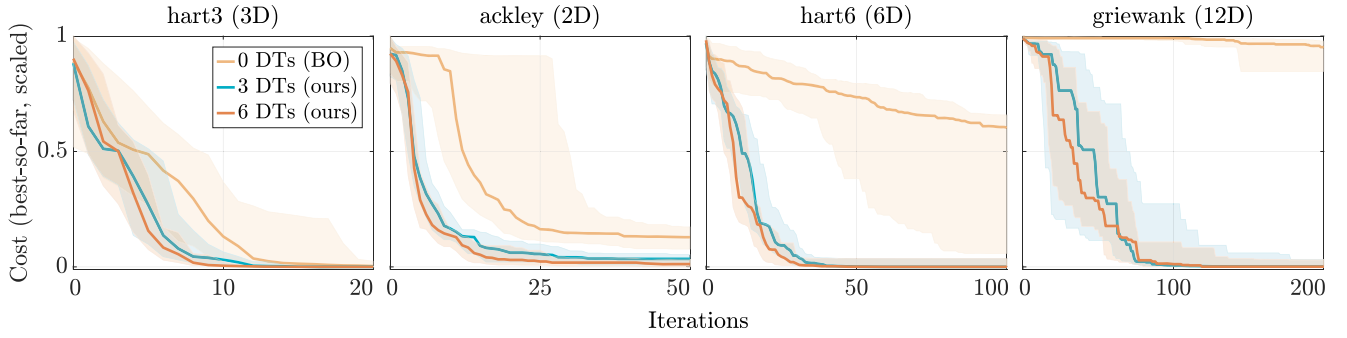


Fig. 5. Convergence rates for the synthetic benchmark functions using standard BO ($P = 0$) and the proposed multi-source BO with $P = 3$ and $P = 6$ DTs. The multi-source BO consistently outperforms standard BO, with larger performance gains for more complex functions (higher dimensionality and more local minima).

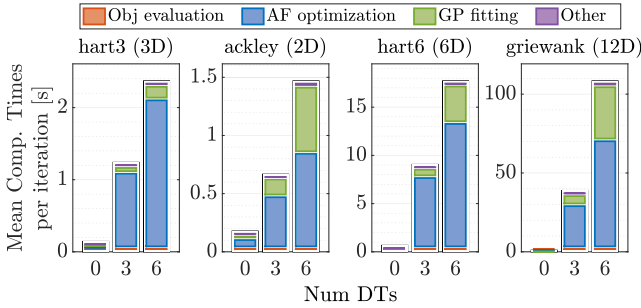


Fig. 6. Average iteration time breakdown across benchmarks. These timings measure software execution in the optimization loop. AF optimization (blue) represents the dominant component compared to GP fitting (green) and objective evaluations (orange).

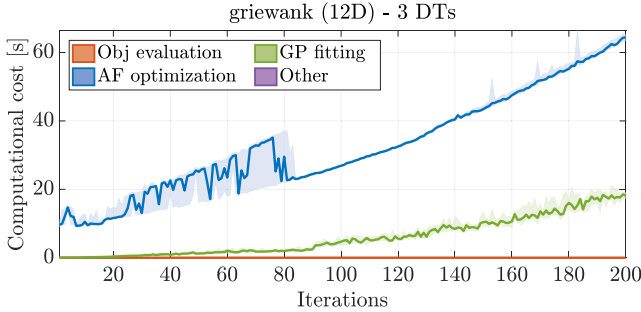


Fig. 7. Computation times over iterations for the Griewank function with $P = 3$ DTs. These timings measure software execution in the optimization loop. The AF optimization and GP fitting times increase with iterations as the dataset grows.

lation study in Section 6.2, where we explicitly test how MSBO behaves when the DT ensemble contains uninformative sources.

6.2. Ablation study on weakly informative sources

In this ablation study, we evaluate the robustness of MSBO when the DT ensemble includes weakly informative sources, namely weakly correlated or uncorrelated DTs. We consider the Ackley function with 3 DTs, where one source is uncorrelated with the RS or where all sources are uncorrelated with the RS. While the RS is the canonical Ackley function, the uncorrelated DTs are generated from perturbed Branin functions. The convergence results in Fig. 10 show that the fully correlated configuration still provides the best performance, as expected. However, introducing one uncorrelated DT does not lead to severe degradation: convergence becomes slightly slower (closer to the BO baseline), though eventually outperforming BO without DTs after approximately 20 iter-

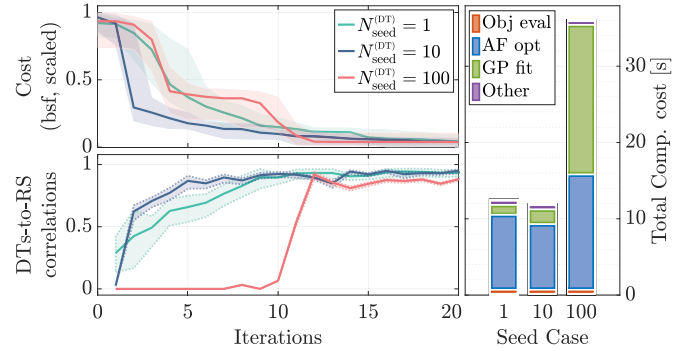


Fig. 8. Effect of offline DT pre-sampling on MSBO (Ackley function, 3 DTs) for three offline DT seed sizes: $N_{\text{seed}}^{(\text{DT})} \in \{1, 10, 100\}$. The top left panel reports convergence times, while the bottom left panel reports the learned DTs-to-RS correlations (the solid curve denotes the mean DTs-to-RS correlation averaged across DTs and 30 runs, the dotted curves denote the maximum and minimum of the same averaged quantity, and the shaded envelope visualizes variability between these bounds). Moderate pre-sampling ($N_{\text{seed}}^{(\text{DT})} = 10$) yields the most consistent early transfer, while large pre-sampling ($N_{\text{seed}}^{(\text{DT})} = 100$) induces an initial correlation lag followed by rapid correlation recovery and superior late-stage convergence. The right panel shows the total algorithmic computation time, which increases with dataset size.

ations. The DTs-to-RS correlation panel clarifies this behavior: in the dark-blue case (one uncorrelated DT), the learned correlations are initially close to zero for all sources; around iteration 20, the ICM separates the sources, keeping one correlation near zero (uninformative DT) while increasing the other two (informative DTs), as reflected by the widening gap between the highest and lowest correlations (dotted curves). This indicates that the model needs an initial adaptation phase to correctly identify informative and uninformative sources; once this identification occurs, MSBO still outperforms BO.

In contrast, when all sources are uncorrelated (pink color), convergence becomes significantly worse, and the ICM correctly assigns near-zero correlation to all DTs, indicating no effective transfer.

Overall, these sensitivity analyses and ablation study demonstrate that MSBO performs best when the DT ensemble is sufficiently informative, moderately sized, and supported by an appropriate amount of offline data. While additional DT data and sources can improve convergence, their benefit is not monotonic: excessive offline samples may introduce a temporary correlation lag, and too many DTs can challenge the ICM's ability to infer reliable inter-source correlations. At the same time, the method remains reasonably robust to a small number of weakly correlated or uncorrelated DTs, since the model can identify and down-weight uninformative sources during optimization. These findings also underline the limitations of the adopted ICM formulation, motivating future work on more flexible multi-source architectures for larger and more heterogeneous digital twin ensembles, as well as on lever-

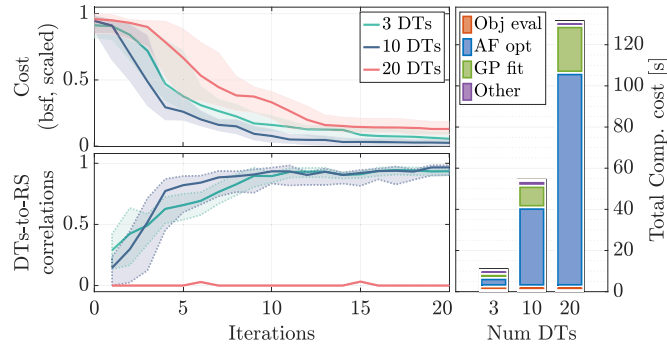


Fig. 9. Effect of DT ensemble size on MSBO (Ackley function) for three different DT ensemble sizes: $P \in \{3, 10, 20\}$. The top left panel reports convergence, the bottom left panel reports the learned DTs-to-RS correlations, and the right panel shows the total algorithmic computation time. Increasing from 3 to 10 DTs slightly strengthens information transfer (correlations) and improves convergence. However, with 20 DTs, the ICM struggles to reliably infer correlations under limited data, leading to near-zero DTs-to-RS correlations and reduced optimization effectiveness.

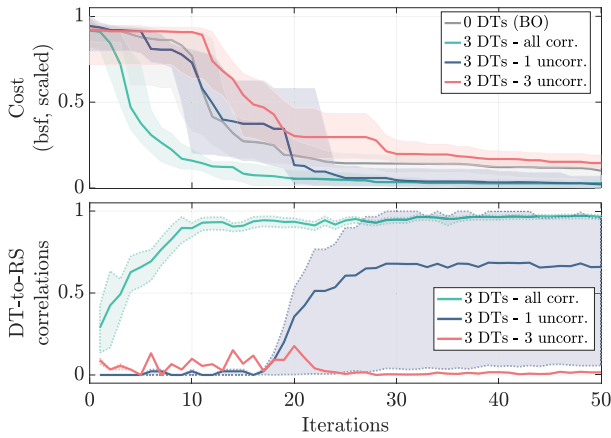


Fig. 10. Impact of one uncorrelated DT on MSBO (Ackley function, 3 DTs). The convergence panel (top) compares BO without DTs, MSBO with 3 correlated DTs, MSBO with 1 uncorrelated DT and MSBO with all 3 DTs uncorrelated. Introducing one uncorrelated DT slightly slows convergence relative to the fully correlated case, but MSBO still outperforms BO without DTs. For this case, the DTs-to-RS correlation of the uninformative source is set to zero while higher correlations are assigned to the remaining DTs (after iteration 20). In contrast, when all sources are uncorrelated (pink color), convergence is significantly worse than all other cases, and the ICM assigns near-zero correlation to all DTs, indicating no effective transfer.

aging the learned correlations to dynamically adapt the DT ensemble during optimization.

7. Experimental results

Following the synthetic validation stage, this section applies the same A-TiL and MSBO pipeline to a physical setup. Specifically, we provide an empirical validation for the speed control of a BLDC motor.

7.1. Experimental setup

The testbed, depicted in Fig. 11, consists of a 100 W BLDC motor rigidly coupled to an aluminum disk which acts as an inertial load ($\approx 1.425 \times 10^{-4} \text{ kg m}^2$), emulating industrial systems like conveyor belts or robotic joints. To address the inherent uncertainty of such loads, we model the system with a fixed but unknown inertia. This motivates the

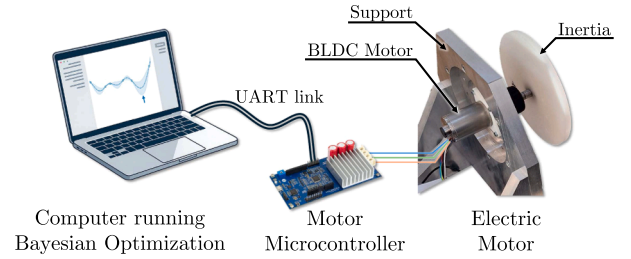


Fig. 11. Experimental testbed setup. The BLDC motor is coupled to an inertial load. The control loop runs on a microcontroller, while parameters are updated online by the A-TiL framework running on an external PC.

use of domain randomization in the DTs, where varying virtual parameters allow the ensemble to capture a range of dynamics and effectively bridge the sim2real gap.

The motor is operated via a dedicated microcontroller, while an external computer runs the MSBO algorithm and the A-TiL parallelized framework and communicates the controller's parameters to the microcontroller. The MSBO hyperparameters are set to $N_{\text{samples}} = 1000$, $N_{\text{raw}} = 250$, $N_{\text{restart}} = 20$, $\beta = 3$, and $N_{\text{seed}}^{(\text{RS})} = N_{\text{seed}}^{(\text{DT})} = 1$.

The control objective is to optimize the PI controller gains $\theta = [k_p, k_i]^T$ within the bounds $k_p \in [10^{-1}, 10^3]$ and $k_i \in [10^{-4}, 1]$. The desired closed-loop behavior is defined by filtering the reference signal through the first-order target model

$$F(s) = \frac{1}{1 + \frac{s}{\omega_c}}, \quad \omega_c = 25 \text{ rad/s}, \quad (14)$$

thus obtaining the desired trajectory y_{des} . The performance metric $J(\theta)$ is defined as a weighted sum of three normalized terms:

$$J(\theta) = J_{\text{tracking}}(\theta) + J_{\text{effort}}(\theta) + J_{\text{smooth}}(\theta), \quad (15)$$

where J_{tracking} is the root-mean-square (RMS) error between the measured motor speed and y_{des} , while J_{effort} and J_{smooth} are the RMS values of, respectively, the control action (current i_q) and its derivative, included to penalize high-frequency chatter and promote smooth control actions.

The Digital Twin. The DT is implemented as a Simulink model capturing the electromechanical dynamics of the system. In practice, the DT ensemble is built in three steps. First, a nominal model is defined using available plant knowledge: electrical motor coefficients from the manufacturer's datasheet and the mechanical load nominal value. Second, we identify which parameters most affect the speed-loop dynamics of interest and should therefore be randomized; parameters with negligible effect in our operating range are not prioritized, while load inertia and air friction are retained as dominant uncertain parameters. Third, we tune the perturbation magnitudes through domain randomization so that the DT ensemble spans plausible plant variability and the real response lies inside the simulated trajectory envelope. In this work, load inertia and air friction are perturbed with Gaussian noise ($\sigma = 10\%$ of nominal identified values).

Fig. 12 provides the first validation level (open-loop consistency): the real motor speed response to current steps is compared against the DT envelope, which broadly contains the measured trajectory. The source correlation panel of Fig. 13 provides a complementary diagnostic during optimization: the learned DTs-to-RS statistical correlations within the ICM surrogate remain non-zero for informative twins, supporting information transfer from DT evaluations to the RS surrogate.

7.2. Optimization results

In this section, we compare the performance of MSBO against standard single-source BO over 20 independent runs, and against a traditional one-shot tuning baseline.

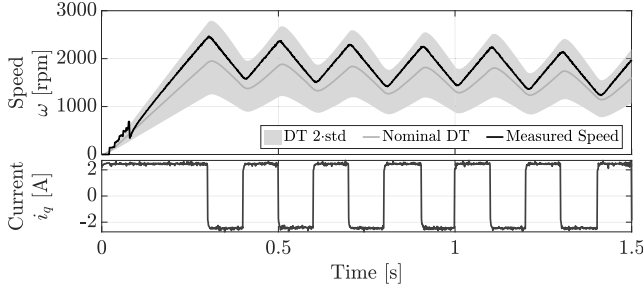


Fig. 12. Open-loop validation of the DT ensemble against the real system. The plot displays the measured speed response (black line) overlaid on the uncertainty generated by the perturbed DTs under current-step excitation. The combined uncertainty (grey) comprehensively covers the real dynamics.

Fig. 13 summarizes the convergence behavior of the BO-based methods. The top left panel reports the scaled cost versus iteration, where the dashed horizontal line denotes the 5% threshold from the optimum. Standard BO (0 DTs) converges more slowly, requiring on average 6 iterations to reach this threshold. In contrast, our proposed multi-source strategy converges in only 2 iterations when using either 3 or 6 DTs. As already observed in the synthetic benchmarks, increasing the number of DTs from 3 to 6 yields only marginal additional benefit in this experimental setting.

The DTs-to-RS correlation panel in the bottom left of Fig. 13 complements this result: informative DTs keep non-negligible learned correlations with the RS throughout the run, which supports model-level cross-source information transfer rather than purely independent parallel search.

The right panel of Fig. 13 reports the corresponding timing breakdown. The dominant contribution to the per-iteration wall-clock time is the on-hardware experiment, which requires approximately 26.5 s per evaluation, whereas the additional software overhead introduced by MSBO remains marginal. Consequently, reducing the average number of iterations from 6 to 2 translates into a decrease in total on-hardware testing time from approximately 159.1 s to 53.0 s.

As an additional benchmark, we compare the obtained result with a one-shot data-driven tuning baseline: Virtual Reference Feedback Tuning (VRFT, Campi et al., 2002). VRFT is a well-known method for direct data-driven controller design, which computes the controller parameters in one shot based on a single open-loop dataset. The method reconstructs the virtual signals associated with a desired target model, which in this experiment is the same model used to define the reference trajectory in Eq. 14. VRFT is attractive in terms of experimental cost, since it requires only a single open-loop dataset and does not involve iterative on-hardware optimization. However, it is limited to linear systems and linear controllers, and in the present experiment its final closed-loop performance is lower than that achieved by the BO-based methods.

A comparison of the closed-loop responses after 3 iterations is shown in Fig. 14. To quantify the control-performance differences visible in this representative multi-step response, additional indicators were computed from the same traces. At iteration 3, MSBO with 3 DTs reduces the root-mean-square error (RMSE) with respect to the filtered reference from 467 rpm for standard BO and 147 rpm for VRFT to 52 rpm. The mean absolute error is similarly reduced from 262 rpm for standard BO and 50 rpm for VRFT to 16 rpm. The maximum overshoot over the multi-step reference transitions decreases from 31% for standard BO and 28.7% for VRFT to 7.3% for MSBO, while the mean first-entry time into the 5% band decreases from 1.56 s for standard BO and 0.29 s for VRFT to 0.13 s for MSBO. These indicators confirm the qualitative behavior shown in Fig. 14: by iteration 3, the MSBO controller tracks the filtered reference more closely, exhibits smaller overshoot after set-point changes, and converges faster to the new operating points than both standard BO at the same iteration and the one-shot VRFT baseline. It is worth

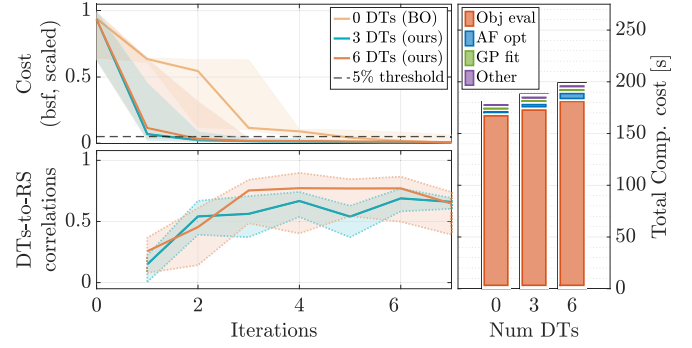


Fig. 13. Experimental results over 20 independent runs. The dashed horizontal line in the convergence plot (top left) marks the 5% threshold from the optimum. MSBO converges significantly faster than standard BO, reaching this threshold on average in 2 iterations instead of 6, thanks to the non-zero DTs-to-RS correlation (bottom left). The right panel shows that the computational overhead introduced by the MSBO algorithm is negligible compared to the physical objective evaluation time of each on-hardware experiment.

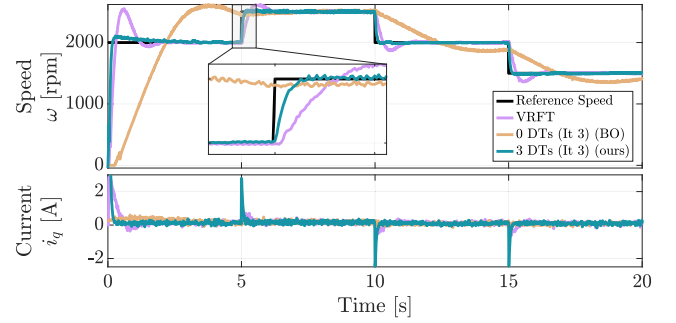


Fig. 14. Representative closed-loop responses at iteration 3. MSBO with 3 DTs (blue) tracks the filtered reference more accurately, exhibits smaller overshoot and transient deviations after set-point changes, and converges faster to the new operating points than both the one-shot VRFT tuning (purple) and standard BO evaluated at iteration 3 (orange). The lower panel shows the corresponding control action i_q . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

noting that the VRFT solution does not correspond to the filtered reference trajectory, since the method is not guaranteed to work perfectly for nonlinear systems and saturated control actions. Table 3 complements this trace-level comparison by reporting the median controller gains and scaled cost over the 20 independent runs. For the BO-based methods, we report both the best-so-far solution at iteration 3 and the best overall solution. The only solutions well below the 5% threshold at iteration 3 are those obtained by MSBO, with a scaled cost of about 1.4% for 3 DTs and 1.8% for 6 DTs, against 6.5% for VRFT and 11.3% for standard BO.

Table 3

Median performance across the 20 independent tests. For each run, the reported BO-based solution is either the best-so-far solution up to iteration 3 or the best on-hardware solution over all iterations. The scaled cost is computed with the same global normalization used in Fig. 13.

Method	Iter.	$\bar{k}_p$	$\bar{k}_i \times 10^{-4}$	$\bar{J}_{\text{scaled}}$
VRFT	1-shot	1.10	0.25	0.065
BO (0 DTs)	3	0.10	1.00	0.113
MSBO (3 DTs)	3	1.30	1.00	0.014
MSBO (6 DTs)	3	2.60	2.90	0.018
BO (0 DTs)	best	1.70	1.00	0.005
MSBO (3 DTs)	best	2.15	2.30	0.005
MSBO (6 DTs)	best	2.72	1.00	0.006

Overall, the proposed method reduces the on-hardware experimental effort and elapsed testing time while also providing high-quality closed-loop performance earlier in the optimization process.

8. Conclusions and future work

In this article, we presented the A-TiL framework, a novel architecture for sample-efficient controller tuning that leverages an ensemble of executable digital twins in parallel with the physical plant. To exploit this architecture, we developed a multi-source Bayesian Optimization algorithm based on a multi-output Gaussian Process ICM. Central to this approach is an innovative batch acquisition function that strategically selects virtual evaluations to minimize model uncertainty within the region of potential improvement on the real system. Numerical benchmarks across various dimensionalities demonstrated that our framework significantly accelerates convergence by effectively bridging the sim2real gap. Finally, experimental validation on a BLDC motor speed control testbed showed that, on average, the proposed strategy reaches near-optimal closed-loop performance using one-third of the on-hardware trials required by standard single-source BO.

Future work will consider more flexible surrogate models for large offline datasets and higher-dimensional or heterogeneous DT ensembles, including multi-fidelity settings.

CRediT authorship contribution statement

Giacomo Delcaro: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Simone Gabrielli:** Writing – review & editing, Visualization, Validation, Software, Data curation; **Simone Formentin:** Supervision, Project administration, Methodology.

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.

Appendix A. Acquisition Function Analysis

As discussed in Section 5.1, batch acquisition functions are often defined using marginalization (Gonzalez et al., 2016; Wilson et al., 2017):

$$a(\theta_1, \dots, \theta_q | D) = \int \alpha(\theta_1, \dots, \theta_q | \mathbf{y}, D) p(\mathbf{y} | D) d\mathbf{y}. \quad (\text{A.1})$$

Here, $\alpha(\theta_1, \dots, \theta_q | \mathbf{y}, D)$ represents the joint utility of the batch conditioned on the hypothetical observation outcomes $\mathbf{y} = [y_1, \dots, y_q]^\top$. In this formulation, α is often defined as the maximum utility among batch candidates relative to a single-point AF (see Table 1 in Wilson et al., 2018). Since $\mathbf{y}$ are unknown a priori, the utility is computed by marginalizing over the GP predictive posterior $p(\mathbf{y} | D)$. However, this integral is typically intractable and requires MC approximations, which significantly increase the cost of the AF optimization process.

The proposed DT AF (Eq. 12) eliminates marginalization over hypothetical observations $\mathbf{y}$ by defining the utility α solely in terms of the

GP posterior variance, which is independent of the observed values. By bypassing the expectation step over $\mathbf{y}$, the complexity is reduced to a spatial integral over R_i . While this integral is also approximated via MC sampling, it requires fewer samples than the marginalization typically used in outcome-dependent batch AFs.

A.1. Role of the integration over R_i

The integral in Eq. (12) is approximated using random test points sampled within the region of interest R_i :

$$a^{(\text{DT})}(\theta_1, \dots, \theta_P | D^*) \approx \sum_{j=1}^{N_{\text{test}}} [\Sigma_{\text{post}}^{(\text{RS})}(\theta_j | D^*) - \Sigma_{\text{post}}^{(\text{RS})}(\theta_j | D^* \cup \{\theta_p^{(\text{DT}_p)}\}_{p=1}^P)], \quad (\text{A.2})$$

where $\{\theta_j\}_{j=1}^{N_{\text{test}}}$ represents a set of N_{test} test points sampled within R_i via Latin Hypercube Sampling, as described in step b of the AF optimization procedure (Section 5). This decoupling of the test locations θ_j from the candidate query points θ_p is critical, as it enables the algorithm to quantify the joint information gain across the entire R_i .

To demonstrate the necessity of integrating over the relevant region R_i , consider a simplified acquisition function that relies solely on point-wise variance reduction:

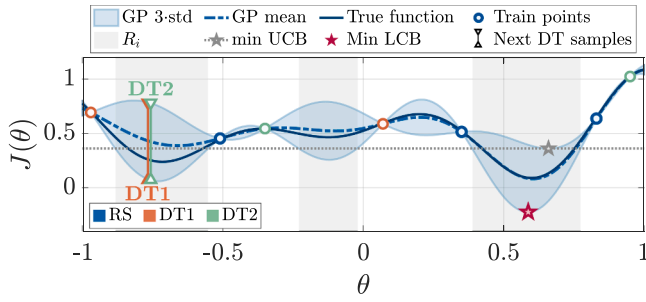
$$a_{\text{point}}^{(\text{DT})}(\theta_1, \dots, \theta_P | D^*) = \sum_{j=1}^P [\Sigma_{\text{post}}^{(\text{RS})}(\theta_p | D^*) - \Sigma_{\text{post}}^{(\text{RS})}(\theta_p | D^* \cup \{\theta_p^{(\text{DT}_p)}\}_{p=1}^P)]. \quad (\text{A.3})$$

In this formulation, the variance reduction is only assessed at the DT query points themselves. Without a fixed set of test points, the optimization objective becomes prone to degeneracy, causing all P candidates to collapse onto the single point of maximum uncertainty. This occurs because the test points vary with the candidates, allowing the optimizer to compute and sum the same local variance reduction P times. Lacking a fixed spatial domain, the optimizer perceives P identical samples as providing P times the information, failing to account for the fact that repeated sampling of a single location yields no additional reduction in uncertainty.

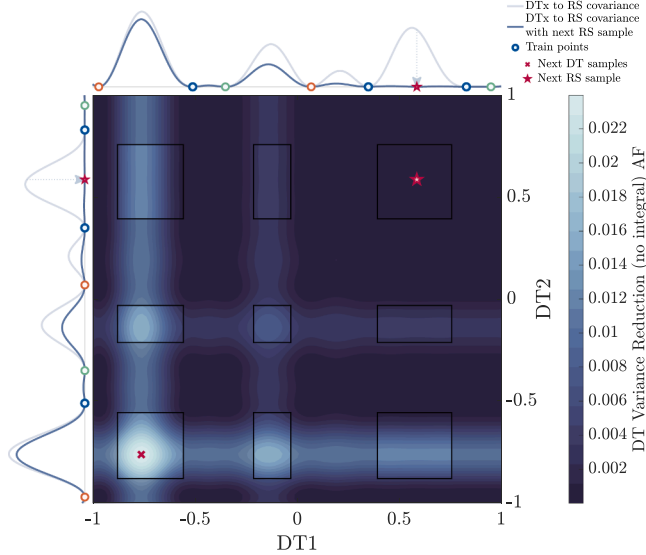
Fig. A.1 illustrates this issue in the perfect correlation scenario described in Fig. 3a. The AF contour plot is substantially different from Fig. 3b: it lacks the valley along the bisector, and the optimization selects identical values for both DTs. By contrast, integrating the variance reduction over R_i serves as a spatial regularizer, ensuring that overlapping queries provide diminishing returns.

A.2. Role of the augmented dataset D^*

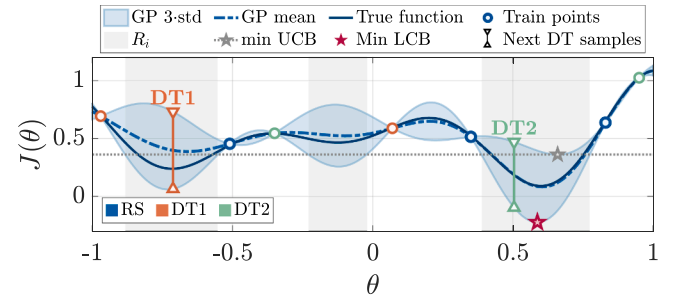
To quantify the impact of the augmented dataset D^* in Eq. (12), Fig. A.2 illustrates the acquisition surface computed using only the existing observations D . Neglecting the impending RS evaluation results in suboptimal DT candidate selection; specifically, one of the AF maxima appears in the immediate vicinity of the RS sample. Because the RS trial will already yield high-fidelity information at this location, such DT queries are redundant. By conditioning the DT selection on D^* , the framework effectively penalizes spatial proximity to the RS candidate, as shown in Fig. 3.



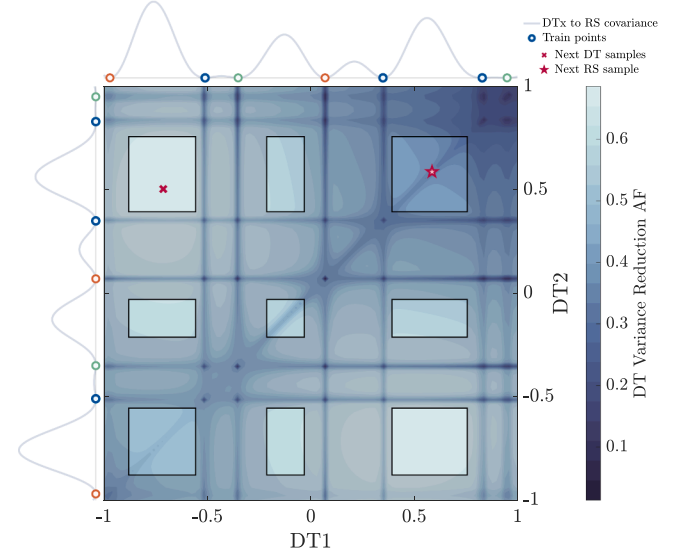
(a) GP posterior for the RS.



(b) Variance reduction acquisition function with no integral term.



(a) GP posterior for the RS.



(b) Variance reduction acquisition function contour plot.

Fig. A.1. Acquisition function in a perfect correlation case, computed without the integral term (Eq. A.3). The AF differs substantially from Fig. 3, and does not exhibit the valley along the bisector. Consequently, the optimization selects identical values for both DTs. This result highlights the critical role of the integral term in preventing redundant sampling.

Fig. A.2. Acquisition function in a perfect correlation case, computed without the RS incumbent in the augmented dataset (i.e., using $\mathcal{D}$ instead of $\mathcal{D}^*$). The DT AF maxima are closer to the next RS sample compared to Fig. 3, leading to suboptimal DT selections.

References

- Allama, J. P., Patrinos, P., Van der Auwera, H., & Son, T. D. (2025). Learning-based NMPC adaptation for autonomous driving using parallelized digital twin. *IEEE Transactions on Control Systems Technology*, 33(4), 1317–1332.
- Alvarez, M. A., Rosasco, L., Lawrence, N. D. et al. (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3), 195–266.
- Bai, T., Li, Y., Shen, Y., Zhang, X., Zhang, W., & Cui, B. (2023). Transfer learning for bayesian optimization: A survey. *arXiv preprint arXiv:2302.05927*
- Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., & Bakshy, E. (2020). BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *Advances in neural information processing systems* 33. <http://arxiv.org/abs/1910.06403>.
- Berkenkamp, F., Schoellig, A. P., & Krause, A. (2016). Safe controller optimization for quadrotors with gaussian processes. In *2016 IEEE International conference on robotics and automation (ICRA)* (pp. 491–496). IEEE.
- Bonilla, E. V., Chai, K., & Williams, C. (2007). Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20, 1–8.
- Busetto, R., Lucchini, A., Formentin, S., & Savaresi, S. M. (2023). Data-driven optimal tuning of BLDC motors with safety constraints: A set membership approach. *IEEE ASME Transactions on Mechatronics*, 28(4), 1975–1983.
- Campi, M. C., Lecchini, A., & Savaresi, S. M. (2002). Virtual reference feedback tuning: A direct method for the design of feedback controllers. *Automatica*, 38(8), 1337–1346.
- Contal, E., Buffoni, D., Robicquet, A., & Vayatis, N. (2013). Parallel gaussian process optimization with upper confidence bound and pure exploration. In *Machine learning and knowledge discovery in databases* (pp. 225–240). Springer Berlin Heidelberg.
- Desautels, T., Krause, A., & Burdick, J. W. (2014). Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(1), 3873–3923.
- Do, B., & Zhang, R. (2025). Multifidelity bayesian optimization: A review. *AIAA Journal*, 63(6), 2286–2322.
- Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Mhs'95. proceedings of the sixth international symposium on micro machine and human science* (pp. 39–43).

- Feurer, M., Letham, B., Hutter, F., & Bakshy, E. (2018). Practical transfer learning for bayesian optimization. *arXiv preprint arXiv:1802.02219*.
- Fröhlich, L. P., Küttel, C., Arcari, E., Hewing, L., Zeilinger, M. N., & Carron, A. (2022). Contextual tuning of model predictive control for autonomous racing. In *2022 IEEE/RSJ International conference on intelligent robots and systems (IROS)* (pp. 10555–10562). IEEE.
- Garces, S. R., De Boi, I., Ramos, J. P., Dierckx, M., Popescu, L., & Derammelaere, S. (2024). Efficient tuning of an isotope separation online system through safe bayesian optimization with simulation-informed gaussian process for the constraints. *Mathematics*, 12(23), 3696.
- Gardner, J. R., Kusner, M. J., Xu, Z. E., Weinberger, K. Q., & Cunningham, J. P. (2014). Bayesian optimization with inequality constraints. In *ICML* (pp. 937–945). (vol. 2014).
- Garnett, R. (2023). Bayesian optimization. Cambridge University Press.
- Gonzalez, J., Dai, Z., Hennig, P., & Lawrence, N. (2016). Batch bayesian optimization via local penalization. In *Proceedings of the 19th international conference on artificial intelligence and statistics* (pp. 648–657). PMLR.
- Grievens, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, pp. 85–113. Cham: Springer International Publishing.
- Jamil, M., & Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150–194.
- Jones, D. R. (2009). Direct global optimization algorithm. In *Encyclopedia of optimization*, second edition (pp. 725–735). Springer.
- Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13, 455–492.
- Kandasamy, K., Dasarthy, G., Oliva, J., Schneider, J., & Poczos, B. (2019). Multi-fidelity gaussian process bandit optimisation. *Journal of Artificial Intelligence Research*, 66, 151–196.
- Kethiri, M. F., Charrouf, O., Betka, A., Tibermacine, I. E., & Napoli, C. (2025). Hybrid fuzzy-PSO based self-tuning fractional order PI controller for BLDC motors in electric vehicles: Comparative analysis and experimental validation. *Journal of Vibration and Control*, 0(0).

- Khosravi, M., Behrunani, V. N., Myszkowski, P., Smith, R. S., Rupenyan, A., & Lygeros, J. (2022). Performance-driven cascade controller tuning with bayesian optimization. *IEEE Transactions on Industrial Electronics*, 69(1), 1032–1042.
- Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC PapersOnLine*, 51(11), 1016–1022. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.
- Larson, J., Menickelly, M., & Wild, S. M. (2019). Derivative-free optimization methods. *Acta Numerica*, 28, 287–404.
- Le Gratiet, L., & Garnier, J. (2014). Recursive co-kriging model for design of computer experiments with multiple levels of fidelity. *International Journal for Uncertainty Quantification*, 4(5).
- Leen, G., Peltonen, J., & Kaski, S. (2012). Focused multi-task learning in a gaussian process framework. *Machine learning*, 89(1), 157–182.
- Liu, H., Cai, J., & Ong, Y.-S. (2018). Remarks on multi-output gaussian process regression. *Knowledge-Based Systems*, 144, 102–121.
- Mahmud, M., Motakabber, S., Alam, A. Z., & Nordin, A. N. (2020). Adaptive PID controller using for speed control of the BLDC motor. In *2020 IEEE International conference on semiconductor electronics (ICSE)* (pp. 168–171). IEEE.
- Makarova, A., Usmanova, I., Bogunovic, I., & Krause, A. (2021). Risk-averse heteroscedastic bayesian optimization. *Advances in Neural Information Processing Systems*, 34, 17235–17245.
- Marco, A., Berkenkamp, F., Hennig, P., Schoellig, A. P., Krause, A., Schaal, S., & Trimpe, S. (2017). Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *2017 IEEE International conference on robotics and automation (ICRA)* (pp. 1557–1563). IEEE.
- Neumann-Brosig, M., Marco, A., Schwarzmann, D., & Trimpe, S. (2019). Data-efficient autotuning with bayesian optimization: An industrial control study. *IEEE Transactions on Control Systems Technology*, 28(3), 730–740.
- Nobar, M., Keller, J., Rupenyan, A., Khosravi, M., & Lygeros, J. (2024). Guided bayesian optimization: Data-efficient controller tuning with digital twin. *IEEE Transactions on Automation Science and Engineering*, 22, 11304–11317.
- Peng, X. B., Andrychowicz, M., Zaremba, W., & Abbeel, P. (2018). Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International conference on robotics and automation (ICRA)* (pp. 3803–3810). IEEE.
- Poloczek, M., Wang, J., & Frazier, P. (2017). Multi-information source optimization. *Advances in neural information processing systems*, 30.
- Saves, P., Diouane, Y., Bartoli, N., Lefebvre, T., & Morlier, J. (2023). A mixed-categorical correlation kernel for gaussian process. *Neurocomputing*, 550, 126472.
- Sepasgozar, S. M. E. (2021). Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment. *Buildings*, 11(4). <https://doi.org/10.3390/buildings11040151>
- Shah, A., & Ghahramani, Z. (2015). Parallel predictive entropy search for batch global optimization of expensive objective functions. *Advances in neural information processing systems*, 28.
- Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: A survey. *Computer*, 27(6), 17–26.
- Sui, Y., Gotovos, A., Burdick, J., & Krause, A. (2015). Safe exploration for optimization with gaussian processes. In *International conference on machine learning* (pp. 997–1005). PMLR.
- Swersky, K., Snoek, J., & Adams, R. P. (2013). Multi-task bayesian optimization. *Advances in neural information processing systems*, 26.
- Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International conference on intelligent robots and systems (IROS)* (pp. 23–30). IEEE.
- Wang, J., Clark, S. C., Liu, E., & Frazier, P. I. (2019). Parallel bayesian global optimization of expensive functions. <https://arxiv.org/abs/1602.05149>.
- Williams, C. K. I., & Rasmussen, C. E. (2006). Gaussian processes for machine learning (vol. 2). MIT press Cambridge, MA.
- Wilson, J., Hutter, F., & Deisenroth, M. (2018). Maximizing acquisition functions for bayesian optimization. In *Advances in neural information processing systems*. Curran Associates, Inc. (vol. 31).
- Wilson, J. T., Moriconi, R., Hutter, F., & Deisenroth, M. P. (2017). The reparameterization trick for acquisition functions. arXiv preprint [arXiv:1712.00424](https://arxiv.org/abs/1712.00424).