



Cite this: DOI: 10.1039/d5dd00572h

# Multi-stage Bayesian optimisation for dynamic decision-making in self-driving labs

Luca Torresi <sup>ab</sup> and Pascal Friederich <sup>\*ab</sup>

Currently, Bayesian optimisation is the most widely used algorithm for identifying informative experiments in self-driving labs (SDLs). While versatile, standard Bayesian optimisation relies on fixed experimental workflows with predefined parameters and objective functions. This prevents on-the-fly adjustments to operation sequences or the inclusion of intermediate results in the decision-making process. Therefore, many real-world experiments need to be adapted and simplified to fit standard SDL settings. In this paper, we introduce multi-stage Bayesian optimisation (MSBO), an extension to Bayesian optimisation that allows flexible sampling of multi-stage workflows and makes data-efficient decisions based on intermediate observables, which we call proxy measurements. MSBO is designed to address common SDL challenges, such as high downstream characterisation costs, sequential dependencies, and the effective use of proxy measurements. To evaluate this approach, we validate our method using computational simulations and retrospective datasets of chemical discovery, demonstrating its potential to accelerate future SDLs. We systematically compare the advantage of taking into account proxy measurements over conventional Bayesian optimisation, in which only the final measurement is observed. We find that across a wide range of scenarios, proxy measurements substantially improve both the time to find solutions and their overall optimality. This not only paves the way to use more complex and thus more realistic experimental workflows in autonomous labs but also to smoothly combine simulations and experiments in the next generation of SDLs.

Received 19th December 2025  
Accepted 26th March 2026

DOI: 10.1039/d5dd00572h

rsc.li/digitaldiscovery

## 1 Introduction

Research in materials science and chemistry has typically relied on intuition and serendipity, and has thus been limited by human cognitive capacity.<sup>1</sup> The traditional process of identifying efficient materials from a range of imaginable compositions usually spans decades. To move beyond these constraints, the scientific community adopted paradigms such as high-throughput methods<sup>2</sup> and design of experiments,<sup>3</sup> ultimately evolving the field into the era of self-driving laboratories (SDLs). These systems integrate robotics and machine learning to enable autonomous experimentation toward human-directed goals without requiring any direct human intervention.<sup>4,5</sup> SDLs perform experiments iteratively, choosing each successive experiment based on previously collected data rather than pre-planning the entire campaign, which maximises the information gain per sample and requires fewer experiments compared to traditional design of experiments.<sup>6</sup>

Several machine learning algorithms have been deployed to drive SDLs, including genetic algorithms,<sup>7</sup> reinforcement

learning,<sup>8-10</sup> and Bayesian optimisation (BO).<sup>11</sup> Among these, BO is the most widely adopted method thanks to its ability to operate effectively with a limited number of evaluations, navigating noisy data and complex landscapes without the need for gradient information. BO has been successfully applied to SDLs for both the optimisation of single objectives,<sup>11-13</sup> and, more recently, for the simultaneous optimisation of multiple objectives.<sup>14</sup>

Despite the considerable advantages these methods offer over traditional approaches, they also come with inherent limitations. A key shortcoming of all these frameworks is their reliance on a fixed experimental workflow. This rigidity prevents making on-the-fly adjustments to the planned sequence of operations. Additionally, standard approaches typically treat experiments as black boxes, ignoring the known structure of the problem, *i.e.* the sequential dependencies between process stages and their intermediate results. As a result, existing frameworks often prove too rigid for practical materials science experiments, which are rarely isolated black-box problems but rather complex, multi-stage processes with tunable parameters at each stage and measurable intermediate outcomes.

One illustrative example is the fabrication of perovskite solar cells, which involves multiple separate stages, such as the deposition of the electrode, transport, and perovskite layers.<sup>15</sup> Although the quality of each layer critically impacts the final

<sup>a</sup>Institute of Nanotechnology, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany. E-mail: pascal.friederich@kit.edu

<sup>b</sup>Institute of Anthropomatics and Robotics, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany



device performance, key metrics such as power conversion efficiency can only be measured after the full device is completed. While it is possible to optimise each layer individually,<sup>16</sup> this approach fails to capture the complex interdependencies between the different components.

### 1.1 Related work

Multi-fidelity Bayesian optimisation,<sup>17,18</sup> also known as multi-information source BO,<sup>19</sup> is a well-established strategy that mitigates the high cost and sample inefficiency of fixed workflows. The approach exploits lower-fidelity approximations of the main objective, which are cheaper to evaluate, may be sampled independently, and are correlated with high-fidelity results. Foundational literature in this domain relies upon multi-level co-kriging models (*e.g.*, Kennedy and O'Hagan,<sup>20</sup> and Forrester *et al.*<sup>21</sup>) to establish autoregressive relationships between fidelities; this approach has subsequently evolved into modern multi-fidelity algorithms, encompassing discrete bandit frameworks such as MF-GP-UCB<sup>22</sup> and continuous BO methods such as in Kandasamy *et al.*<sup>23</sup>. These methods model evaluations at different fidelities jointly (*e.g.*, *via* multi-output Gaussian processes) and strategically balance low-cost approximate measurements with high-cost accurate measurements. Selection strategies either decouple the choice of fidelity and query location in the input parameter space or incorporate evaluation cost directly into the acquisition function to decide both simultaneously. However, multi-fidelity BO still inherits key limitations of standard BO, including reliance on fixed workflows, limited use of intermediate measurements, and inadequate handling of structured multi-stage processes. Furthermore, these methodologies query a singular objective at varying levels of precision; they assume that the low-fidelity proxy and the final objective are merely correlated approximations of the same underlying target. Consequently, they fail to model causal cascades composed of structurally distinct, heterogeneous experimental stages where intermediate outputs serve as inputs to subsequent operations.

Going beyond multi-fidelity BO, Astudillo and Frazier<sup>24</sup> extended BO to structured objectives by considering composite functions of the form  $f(x) = g(h(x))$ , where  $h$  is an expensive vector-valued black box and  $g$  is a cheap real-valued function. They modelled  $h$  with a Gaussian process (GP) and designed a Monte Carlo-based acquisition function, called expected improvement for composite functions, which uses intermediate measurements to guide the optimisation process. This framework was later generalised to directed acyclic graphs (DAGs) of functions,<sup>25</sup> where each node depends on its parents and is modelled as an independent GP. This approach, termed Bayesian optimisation of function networks (BOFN), leverages known structure for more efficient optimisation but requires evaluating the full network at each iteration, limiting flexibility in selectively sampling sub-graphs.

Recent extensions, such as partially observable knowledge-gradient methods for function networks,<sup>26</sup> relax the rigidity of standard BO by allowing selective evaluation of intermediate nodes. While the original formulation achieves strong query

efficiency, it comes at a high computational cost due to its nested Monte Carlo estimators and repeated acquisition optimisations. A faster variant<sup>27</sup> reduces runtime by optimising over subsets of the input space, with modest reductions in solution quality. Both approaches assume that intermediate nodes can be probed at arbitrary inputs, independent of upstream stages. While this abstraction facilitates algorithm design, it may not hold in practical laboratory settings, where intermediate measurements usually require executing preceding stages of the process, and the possible value ranges of observations at each stage are typically not known in advance.

Building on cascade Bayesian optimisation<sup>28</sup> and BOFN, Kusakawa *et al.*<sup>29</sup> proposed a method for cascade processes, modelling each stage as a separate GP and dynamically adjusting downstream stage parameters conditional on the measured outputs of earlier stages, with the core contribution focusing on this continuous, multi-stage look-ahead sampling. The effectiveness of the method was demonstrated using a combination of synthetic numerical test functions and a solar cell simulator learned from data.

An alternative paradigm for sample efficiency relies upon dynamic resource allocation and early-stopping mechanisms; notable examples include Hyperband,<sup>30</sup> BO hyperband,<sup>31</sup> and Freeze-Thaw BO,<sup>32</sup> which have become established as state-of-the-art methodologies for hyperparameter optimisation in neural networks. However, because such algorithms rely upon a temporal continuation assumption for a singular process (for instance, by extrapolating asymptotic learning curves as in Freeze-Thaw), they cannot model the complex transfer functions required when intermediate observables transition between structurally distinct, heterogeneous physical domains. Furthermore, approaches such as Hyperband employ deterministic pruning heuristics intended for massive parallelisation; such mechanisms prove fundamentally incompatible with the costly, sequential experimentation inherent to self-driving laboratories.

### 1.2 Contributions

In this work, we develop a multi-stage Bayesian optimisation framework (MSBO), which integrates proxy and intermediate-stage measurements into BO. MSBO diverges from existing network-based approaches, such as BOFN, in three fundamental respects: it permits partial evaluations of the cascade rather than enforcing monolithic, full-network iterations; it employs a state-saving inventory system to pause and resume partially processed samples; and, finally, it utilises a nested, stage-aware acquisition function to direct sampling toward specific, causally feasible stages. By contrast, standard BOFN applies a global acquisition function across the entire workflow, and variants permitting selective evaluation assume intermediate nodes can be probed independently, thereby violating the strict causal dependencies of experimental cascades. This architecture enables resumable sampling and flexible, on-the-fly workflow adaptation across structurally distinct operations, thereby reducing optimisation costs. We design a synthetic function generator for multi-stage optimisation tasks,



supporting systematic exploration of varying stage complexities, correlations, and levels of information loss through additive noise and masking. The function generator aims to mimic real-world applications' complexity better than standard analytical test functions used frequently to benchmark BO algorithms. We extensively evaluate MSBO, showing that integrating a flexible acquisition function with a multi-stage GP surrogate consistently outperforms standard BO across diverse synthetic benchmarks and real-world problems. We furthermore quantify the efficiency gains in various relevant self-driving lab scenarios involving two-stage processes with varying complexities and cost ratios between the first proxy measurement and the second, more expensive measurement of the objective. We demonstrate that even for the combination of very simple proxy measurements and very complex objective measurements, the multi-stage Bayesian optimisation approach outperforms standard BO, paving the way to a broader use of proxy measurements in all self-driving lab settings, and also to including (cheap and approximative) computational steps in experimental optimisation workflows.

## 2 Preliminaries

### 2.1 Bayesian optimisation

Bayesian optimisation is an effective framework for finding the global maximum of an expensive black-box function  $f(\mathbf{x})$  over a bounded domain  $\mathcal{D}$ :

$$\max_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) \quad (1)$$

BO is particularly suited for functions where evaluations are costly, noisy, or lack an analytic expression, precluding the use of gradient-based methods.<sup>33</sup> The core idea is to balance exploration, *i.e.* sampling regions with high uncertainty, and exploitation, *i.e.* sampling regions likely to improve the current best-observed value, to reach the global optimum in a minimal number of function evaluations.

The BO process is iterative, relying on two essential components: a surrogate model and an acquisition function. The surrogate model is a probabilistic model, constructed using a small set of observations to approximate  $f(\mathbf{x})$ . It provides fast estimates of predictions and associated uncertainty across the optimisation domain  $\mathcal{D}$ . The acquisition function quantifies the expected utility of evaluating  $f$  at a specific point  $\mathbf{x}$ . The next sampling location  $\mathbf{x}_{t+1}$  is chosen by maximising this computationally inexpensive acquisition function. The alternating execution of these two steps efficiently generates a sequence of samples converging to the global optimum.

**2.1.1 Gaussian processes.** Gaussian processes are a common choice for the BO surrogate model due to their non-parametric nature and reliable, built-in uncertainty quantification.<sup>34</sup> A GP is a collection of random variables, any finite number of which have a joint multivariate Gaussian distribution.<sup>35</sup> It defines a prior distribution over functions  $f: \mathcal{D} \rightarrow \mathbb{R}$ , specified completely by a mean function  $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$  and a covariance function  $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$ :

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \quad (2)$$

The covariance function, or kernel  $k(\cdot, \cdot)$ , is a positive semi-definite function (*e.g.*, Matern or radial basis function kernels) that defines the similarity between input pairs  $\mathbf{x}$  and  $\mathbf{x}'$ . The mean function is often taken to be zero for simplicity.

While standard GP training suffers from  $\mathcal{O}(N^3)$  computational complexity, various approximation methods exist to scale GPs to large datasets.<sup>36</sup> Furthermore, GPs can be extended to model multiple correlated outputs using multi-task Gaussian processes,<sup>37</sup> often employing an intrinsic coregionalization model to define the inter-task covariance structure.<sup>38</sup>

**2.1.2 Acquisition functions.** Acquisition functions  $\mathcal{L}(\mathbf{x})$  translate the probabilistic belief from the surrogate model into a metric for selecting the next query point. Maximising  $\mathcal{L}(\mathbf{x})$  replaces the original expensive optimisation problem with a computationally cheap one.

Acquisition functions are typically defined as the expectation of a utility function  $l(y)$ , which measures the desirability of a hypothetical outcome  $y$  given the current input  $\mathbf{x}$ . If  $p(y|\mathbf{x})$  is the predictive distribution from the surrogate model,  $\mathcal{L}(\mathbf{x})$  is defined as:

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}[l(y)] = \int l(y)p(y|\mathbf{x})dy \quad (3)$$

This integral is often approximated using methods such as Monte Carlo estimation. Standard choices for acquisition functions include the expected improvement (EI) and the upper confidence bound.<sup>39</sup>

### 2.2 Expected improvement

The EI represents the average amount by which sampling a point  $\mathbf{x}$  is expected to improve upon the current best-observed objective value. The improvement is defined by the utility function  $l(y) = \max(0, y - f_{\max})$ . For a GP with predictive mean  $\mu(\mathbf{x})$  and standard deviation  $\sigma(\mathbf{x})$ , the EI can be computed analytically as:

$$\mathcal{L}_{\text{EI}}(\mathbf{x}) = \mathbb{E}[\max(0, y - f_{\max})] = \sigma(\mathbf{x})[Z\Phi(Z) + \phi(Z)] \quad (4)$$

where  $Z = \frac{\mu(\mathbf{x}) - f_{\max}}{\sigma(\mathbf{x})}$ ,  $\Phi(\cdot)$  is the cumulative distribution function of the standard normal distribution, and  $\phi(\cdot)$  is the probability density function of the standard normal distribution.

### 2.3 Cascade processes

Many complex systems, in particular in materials science and chemistry, but also in optimisation problems beyond these disciplines, are structured as multistage processes, where a sequence of operations must be executed to achieve a final result. Examples are materials characterisation steps before thin-film property measurements or even device measurements; *in situ* measurements of molecules taken during synthesis or after purification, followed by more complex measurements of molecular functionality; and generally any computations or



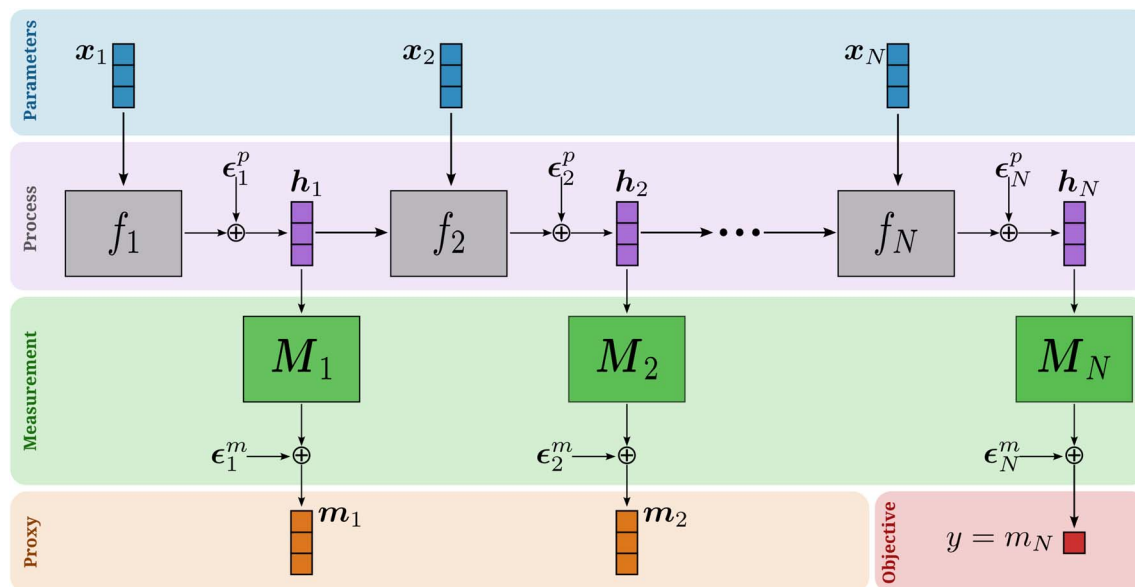


Fig. 1 Illustration of a cascade process of  $N$  stages. At each stage ( $i$ ), the process function  $f_i$  takes controllable parameters  $x_i$  and the latent output from the previous stage  $h_{i-1}$  to generate a new latent state  $h_i$ , subject to process noise  $\epsilon_i^p$ . The optimizer does not observe  $h_i$  directly but receives a proxy measurement  $m_i$ , obtained via a masking operator  $M_i$  and additive measurement noise  $\epsilon_i^m$ . The final objective is defined as the terminal measurement  $y = m_N$ .

simulations preceding experimental synthesis during molecular and materials design.

We define a cascade process as a specific type of multistage system presenting a sequential and dependent structure (see Fig. 1). In a cascade process, the product, or output, ( $h_{i-1}$ ) of any given stage serves as input for the subsequent stage ( $i$ ). Formally, a cascade process consists of  $N$  stages, where each stage  $i$  is represented by a function  $f_i$ . The stage function depends on a set of controllable parameters  $x_i$  and the output from the preceding stage:

$$h_i = f_i(x_i, h_{i-1}) + \epsilon_i^p, \text{ for } i = 1, \dots, N. \quad (5)$$

where  $\epsilon_i^p$  is noise associated with the process. The final system output is  $y = h_N$ . This structure is common in areas like chemical manufacturing, semiconductor fabrication, and material processing, where the outcome of an initial reaction or preparation step directly constrains and influences all subsequent steps. This dependency means that sub-optimal performance at an early stage propagates through the entire system, making the joint optimisation of all controllable parameters  $\{x_i\}_{i=1}^N$  a challenging problem.

### 3 Method

We present MSBO (multi-stage Bayesian optimisation), a Bayesian optimisation framework designed for sequential multi-stage processes. MSBO addresses the challenge of optimising complex chains where the output of each operation serves as the input for subsequent stages. The framework models the system's dependency structure by employing a cascade of GPs inspired by Astudillo and Frazier,<sup>25</sup> which is

then paired with nested acquisition function evaluations to account for heterogeneous computational costs and parameter spaces.

#### 3.1 Optimisation objective

Consider a cascade process, as defined in Section 2.2 and illustrated in Fig. 1, consisting of  $N$  distinct operations. Let  $f_i$  denote the true (latent) function governing the  $i$ -th stage, where the intermediate output  $h_i$  is a function of the tunable inputs  $x_i$  and the preceding latent output  $h_{i-1}$ . The key distinction for optimisation is that the true outputs  $h_i$  are not always directly observable. Instead, we obtain a measurement  $m_i$  of the intermediate output  $h_i$  at stage  $i$ , defined as:

$$m_i = M_i h_i + \epsilon_i^m \quad (6)$$

where  $M_i$  is a (binary diagonal masking) matrix that selects a subset of the state  $h_i$  for observation (to model partially observable states), and  $\epsilon_i^m$  is the additive observation noise. A generalisation to arbitrary matrices  $M$  is straightforward, but not tested in this work. The overall objective is to maximise the final measured output  $y = m_N$  through the optimal selection of the joint parameter set  $\mathcal{X} = \{x_1, \dots, x_N\}$ , where we assume the final measurement is scalar. While we restrict this study to scalar objectives, the framework can be readily extended to multi-objective BO.

#### 3.2 Resumable sampling

A core feature of our framework is its capability to handle partially processed samples through the use of an inventory. Unlike conventional optimisation approaches that require the



complete evaluation of the final objective function, MSBO permits stopping and resuming sample processing at any intermediate stage of the cascade. The inventory maintains a record of all sample states, process parameters, and measurements, tracking which sub-processes have been completed for each sample ID.

The use of an inventory allows for sample continuation: the optimisation can resume from any intermediate state, and partially processed sample states can be retrieved and applied as inputs to subsequent sub-processes. This system promotes data efficiency, since intermediate results are preserved and reused rather than being discarded, thereby avoiding redundant computations. Finally, this architecture supports dynamic resource allocation by allowing the algorithm to select which sub-process to sample next based on acquisition function values.

This resumable sampling capability allows the algorithm to dynamically balance effort between advancing existing samples with high expected utility through subsequent evaluation stages *versus* initiating new samples to explore other regions of the parameter space. The resulting framework is particularly valuable in real-world scenarios, enabling dynamic decision-making and enhancing overall data efficiency.

### 3.3 Cascade Gaussian processes

To capture the sequential nature of the data, we model the system using a cascade of independent GPs (see Fig. 2). Each stage  $i$  is approximated by a probabilistic surrogate:

$$f_i(\mathbf{x}_i^{\text{aug}}) \sim \text{GP}(\mu_i(\mathbf{x}_i^{\text{aug}}), k_i(\mathbf{x}_i^{\text{aug}}, \mathbf{x}_j^{\text{aug}})) \quad (7)$$

where  $\mathbf{x}_i^{\text{aug}}$  represents the augmented input vector. The framework supports two architectural configurations:

- Standard cascade:  $\mathbf{x}_i^{\text{aug}} = [\mathbf{x}_i, \mathbf{m}_{i-1}]$ , creating a direct Markovian dependency.
- Residual connections:  $\mathbf{x}_i^{\text{aug}} = [\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{m}_{i-1}]$ , enabling the model to preserve information from previous inputs, analogous to residual links in deep neural networks. This configuration helps mitigate information loss, especially with partially observed outputs.

To propagate the epistemic uncertainty (the uncertainty in the model due to the limited amount of training data) of the

GPs along the chain, we implement a Monte Carlo sampling approach. For a cascade of  $N$  processes, uncertainty propagation proceeds as follows: at stage  $i = 1$ , we draw  $S$  samples from the posterior of the first GP,  $\mathbf{m}_1^{(s)} \sim \text{GP}_1(\mathbf{x}_1)$  for  $s = 1, \dots, S$ . For subsequent stages  $i > 1$ , each sample  $\mathbf{m}_{i-1}^{(s)}$  from the previous process serves as an input to the next GP, yielding  $\mathbf{m}_i^{(s)} \sim \text{GP}_i([\mathbf{x}_i, \mathbf{m}_{i-1}^{(s)}])$ . This sampling-based approach naturally captures the non-Gaussian, potentially multimodal uncertainty distributions that emerge from the cascade of nonlinear transformations, providing realistic uncertainty quantification for the final process outputs. The resulting ensemble of samples  $\{\mathbf{m}_K^{(s)}\}_{s=1}^S$  represents the compound uncertainty and enables acquisition functions to make decisions that account for uncertainty accumulation throughout the entire process cascade.

### 3.4 Multi-stage expected improvement

To fully leverage the resumable sampling capability described earlier, we propose a nested optimisation strategy. This approach constructs an acquisition function that takes into account the sequential dependency of the cascade process, where the utility of a decision at an early stage depends on the potential outcomes of subsequent stages.

We formulate the acquisition function recursively. For the final stage  $N$ , the acquisition function is the standard EI on the objective  $y$ , conditioned on the previous measurement  $\mathbf{m}_{N-1}$ . For any preceding stage  $i < N$ , the acquisition function is defined as the expected value of the acquisition function of the subsequent stage  $i + 1$ , integrated over the probabilistic outcome of the current stage  $\mathbf{m}_i$ , thus allowing the uncertainty at intermediate stages to propagate forward to the final objective.

Formally, let  $y^*$  denote the best objective value observed so far. The nested acquisition function  $\alpha$  is defined recursively as:

$$\begin{cases} \alpha_{\text{EI}|\mathbf{m}_{N-1}}(\mathbf{x}_N) = \mathbb{E}_{y \sim \text{GP}_N(\mathbf{x}_N^{\text{aug}})} [\max(0, y - y^*)] \\ \alpha_{\text{EI}|\mathbf{m}_{N-2}}(\mathbf{x}_{N-1}, \mathbf{x}_N) = \mathbb{E}_{\mathbf{m}_{N-1} \sim \text{GP}_{N-1}(\mathbf{x}_{N-1}^{\text{aug}})} [\alpha_{\text{EI}|\mathbf{m}_{N-1}}(\mathbf{x}_N)] \\ \vdots \\ \alpha_{\text{EI}}(\mathbf{x}_1, \dots, \mathbf{x}_N) = \mathbb{E}_{\mathbf{m}_1 \sim \text{GP}_1(\mathbf{x}_1)} [\alpha_{\text{EI}|\mathbf{m}_1}(\mathbf{x}_2, \dots, \mathbf{x}_N)] \end{cases} \quad (8)$$

At each iteration, the optimiser queries the inventory to identify all valid continuation points, as well as the option to

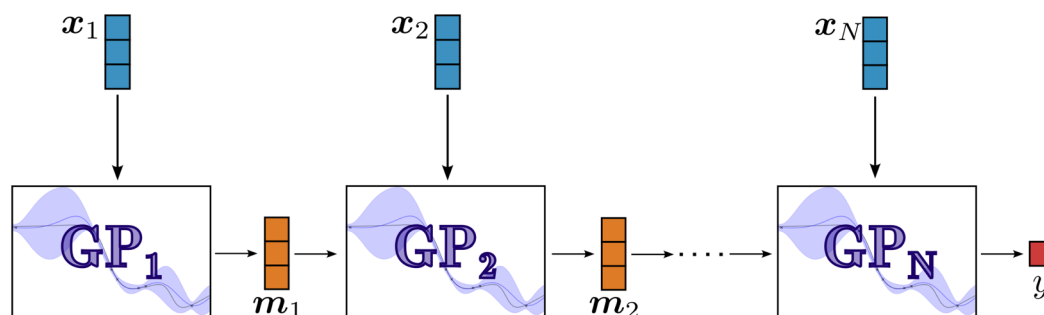


Fig. 2 Schematic illustration of the standard cascade configuration of our surrogate model as a chain of independent GPs. The probabilistic output distribution of the upstream model  $\text{GP}_{i-1}$  serves as a noisy input for the downstream model  $\text{GP}_i$ . Epistemic uncertainty is propagated through the chain, from the initial parameters  $\mathbf{x}_1$  to the final objective  $y$ , via Monte Carlo sampling.



initialise a new sample at the first sub-process. For each candidate process  $i$  and its corresponding available input  $\mathbf{m}_{i-1}$  (retrieved from the inventory), we optimise the parameters  $\mathbf{x}_i$  to maximise the nested acquisition function. To prevent stagnation in regions where the EI vanishes (*i.e.*, is below a numerical tolerance of  $5 \times 10^{-3}$  across the search space), the algorithm automatically switches to the upper confidence bound acquisition function as a tie-breaking mechanism. The associated exploration weight,  $\beta$ , follows a linear decay schedule (from an initial value of 100 to 1) as a function of the remaining budget to favour exploration.

The final selection of the next experiment is determined by comparing the maximal acquisition values across all stages. The framework allows for cost-weighted selection to account for heterogeneous experimental costs  $c_i$  per stage:

$$\text{Next stage} = \underset{i, \mathbf{x}_i}{\operatorname{argmax}} \frac{\alpha(\dots)}{c_i}. \quad (9)$$

To ensure numerical stability, we enforce strictly positive costs ( $c_i > 0$ ). For stages with physically negligible costs, a small positive constant (*e.g.*,  $\varepsilon = 10^{-6}$ ) must be assigned to avoid singularities. In the experiments reported here, setting a uniform cost ( $c_i = 1$ ) for all stages yielded superior optimisation performance compared to explicitly modelling the cost ratios; therefore, we use uniform costs unless otherwise stated. We hypothesise that strict cost-weighting may inadvertently penalise expensive but highly informative regions. In contrast, a uniform cost structure ensures the algorithm prioritises global information gain and solution quality over short-term cost savings. The complete algorithmic procedure governing stage selection and the corresponding inventory updates is detailed in Algorithm 1 of the SI.

### 3.5 Sampling strategy

The experimental design strategy consists of two phases: First, the acquisition of an initial dataset  $\mathcal{D}_0$  through quasi-random sampling of the entire cascade to build an initial surrogate model. To this aim, we use Sobol sequences, ensuring space-filling properties in the high-dimensional joint input space. In all experiments, the size of  $\mathcal{D}_0$  was set to  $2(D + 1)$ , where  $D$  is the total number of controllable parameters (components of all  $\mathbf{x}_i$  for  $i = 1, \dots, N$ ). This is followed by an adaptive optimisation phase, where new sampling points are selected based on the acquisition function. In this adaptive phase, the algorithm can choose from which sub-process to sample next, rather than being restricted to evaluating the entire cascade from the initial input. We allow users to define a constraint on the minimal relative frequency at which each sub-process is sampled. This constraint is essential in settings where the final sub-process is relatively easy to model. In this scenario, the high uncertainty propagating from complex upstream stages can dominate the acquisition function, leading the algorithm to undersample the final stage and consequently fail to realise the optimal set of parameters (*i.e.*, the best objective value).

### 3.6 Implementation details

All GPs use scaled radial basis function kernels with automatic relevance determination. All inputs are normalised to  $[0, 1]^d$ , and outputs are standardised to zero mean and unit variance. Hyperparameters are optimised *via* marginal likelihood maximisation at each iteration. The Monte Carlo integration for uncertainty propagation utilises a Sobol quasi-random sequence with  $S = 256$  samples. The acquisition function optimisation employs a multi-start gradient-based approach (L-BFGS-B) with Sobol initialisation to ensure global search capability. For experiments involving discrete input spaces, the acquisition function is instead computed over all possible inputs, and the maximum is selected *via* exhaustive search.

## 4 Experiments

To validate our MSBO framework, we perform a comprehensive evaluation across synthetic functions and real-world molecular property prediction tasks. We demonstrate the superior data efficiency and robustness of MSBO, arising from its ability to explicitly model cascade processes and dynamically manage intermediate measurements, compared to standard methodologies. We ablate the contributions that stem from the cascade GP and the flexible multi-stage acquisition function with the resumable sampling mechanism.

### 4.1 Experimental setup

**4.1.1 Baselines.** We compare MSBO against three baselines:

- **Random search:** in each iteration, the set of candidate parameters  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$  is drawn from a uniform distribution. This establishes a lower bound for optimisation performance.
- **Standard BO:** this baseline treats the entire multi-stage process as a single black-box function, optimising the final objective  $y(\mathbf{x}_1, \dots, \mathbf{x}_N)$  without leveraging intermediate measurements. It uses the expected improvement acquisition function.
- **BOFN:<sup>25</sup>** this baseline leverages the known function network structure and intermediate measurements *via* the same cascade GP surrogate model employed by MSBO. However, it operates under a monolithic evaluation scheme, requiring the execution of all stages per iteration to obtain the final objective. Consequently, it neither maintains an inventory for resumable sampling nor employs a stage-aware acquisition rule; rather, it applies a standard expected improvement acquisition function across the entire network.

**4.1.2 Metrics.** Our primary metric for evaluating performance is the logarithm of the simple regret,

$$R_t = \log(|y_{\text{opt}} - y_t^*|),$$

where  $y_{\text{opt}}$  is the global optimum of the objective function and  $y_t^*$  is the best objective value observed up to iteration  $t$ . In settings with process noise, noisy samples may occasionally yield values exceeding the true, denoised optimum. To address this, we define the reference  $y_{\text{opt}}$  as the true optimum plus three



times the standard deviation of the cumulative output noise. This prevents the regret metric from becoming undefined or artificially saturated due to stochastic outliers. For the real-world task, we report percentile-based measures that reflect the best-performing fraction of the dataset discovered by the optimiser.

**4.1.3 Heterogeneous costs.** Unless otherwise specified, we adopt a uniform cost model in which generating a sample for any sub-process (*i.e.*, completing any stage  $i$ ) consumes  $c_i = 1$  unit of budget. This allows a direct comparison based on sampling efficiency. We also consider heterogeneous cost scenarios to simulate more expensive downstream stages (Fig. 5 and 6). In those experiments, we normalise the total cost of sampling the whole cascade to 1 so that each full evaluation consumes the same total budget, but the cost is fractioned differently between sub-processes; this normalisation simplifies visual comparison of optimisation trajectories under different cost repartitions.

## 4.2 Synthetic benchmarks

We benchmark MSBO using our synthetic function generator (details in SI A), which provides fine-grained control over function complexity, input dimensionality, and noise properties, thereby creating a controlled environment for evaluating our framework. The generator constructs multi-stage processes by chaining interconnected neural networks, where each stage is trained on a random seed dataset to create continuous, differentiable functions. By varying the size of these seed datasets, we can implicitly tune the complexity of each sub-process while simulating realistic constraints such as information masking and error propagation.

**4.2.1 Conceptual demonstration of MSBO sampling efficiency.** We first examine a simple two-stage cascade  $f_1(x_1) \rightarrow f_2(h_1, x_2) \rightarrow y$ , with both  $x_1$  and  $x_2 \in \mathbb{R}$ , so that both landscapes can be visualised directly. In these experiments, we set  $\mathbf{M} = \mathbf{I}$  and both process and measurement noise to 0. The complexity of the first and second stages is set to 8 and 2, respectively. In our framework, complexity is controlled by the number of random seed points used to construct the synthetic functions, where higher values correspond to more rugged landscapes and increased modelling difficulty of the objective (see SI A).

We find that, for any given cost, MSBO consistently achieves lower regret than the standard BO baseline and random search, demonstrating improved data efficiency (see Fig. 3 for representative results from these experiments). All experiments are averaged over 10 independent runs to obtain uncertainty bounds and to ensure statistical significance. Furthermore, while MSBO suggests a similar number of first-stage function evaluations as BO (left panels in Fig. 3B and C), it allocates second-stage resources more efficiently, concentrating samples in the region of the global optimum. In the standard BO run, the samples for the second process are broadly scattered across the domain and are not concentrated near the true optimum of the second-stage landscape. By contrast, MSBO concentrates the majority of its second-stage samples in the region close to the maximum of the second-stage objective, showing that the

method is substantially more efficient at allocating second-stage evaluations to informative regions.

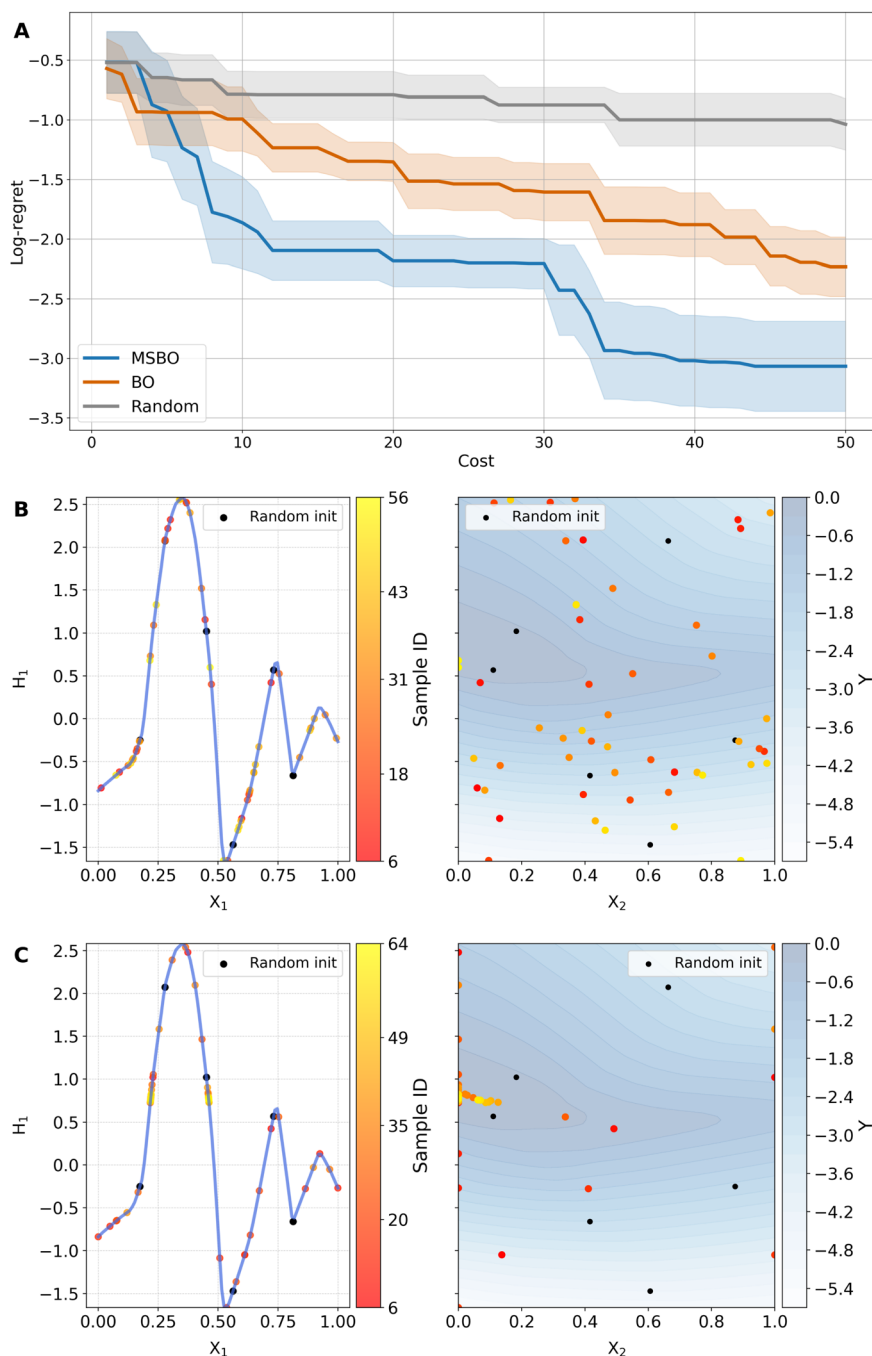
MSBO discovers and explores multiple high-performing modes in the first process: samples are clustered around several distinct regions of the  $f_1$  landscape that correspond to high-valued objectives, indicating the algorithm's ability to identify and maintain promising alternatives rather than collapsing prematurely to a single local mode. This behaviour is a direct consequence of (i) the cascade GP that propagates multi-modal uncertainty, and (ii) the nested, resumable acquisition strategy that allows the optimiser to selectively invest further budget only for the most promising samples, pruning the search space early. More examples are provided in SI B (Fig. S7 and S8).

**4.2.2 Complexity sweep.** We now assess the behaviour of MSBO on more challenging synthetic tasks where the two-stage cascade process is defined as  $f_1(x_1) \rightarrow f_2(\mathbf{h}_1, \mathbf{x}_2) \rightarrow y$ . In this setting, the control parameters are  $\mathbf{x}_1 \in \mathbb{R}^4$ , the intermediate observables are  $\mathbf{h}_1 \in \mathbb{R}^2$ , and the second-stage control parameters are  $\mathbf{x}_2 \in \mathbb{R}^2$ . We assume fully observable states ( $\mathbf{M} = \mathbf{I}$ ) and no noise. The synthetic generator allows us to independently adjust the complexity of both the first ( $f_1$ ) and second ( $f_2$ ) sub-processes (see SI A for details). This enables a broad exploration across different relative complexity settings.

To explore different possible real-world scenarios, our experiments test nine combinations of first- and second-stage complexities. We compare MSBO with standard BO (which treats the cascade as a monolithic black box) and the BOFN baseline, which can also use the intermediate measurement  $\mathbf{m}_1 = \mathbf{h}_1$ . We find that MSBO consistently outperforms the other baselines in all scenarios. The ability of MSBO and BOFN to learn the intermediate mapping  $\hat{h}_1(x_1)$  provides a strong driving signal, overcoming the difficulties standard BO faces in optimising over the high-dimensional joint space  $(\mathbf{x}_1, \mathbf{x}_2)$ . This is clearly demonstrated by the significant performance gap that both MSBO and BOFN exhibit with respect to BO across the entire sweep. Furthermore, MSBO's multi-stage acquisition function and support for resumable sampling provide a substantial, ulterior boost in performance over all baselines, including BOFN. MSBO consistently achieves the lowest regret for a given cost and maintains a clear margin over the baselines. This advantage persists whether the difficulty arises primarily in the first stage (left column), in the second stage (bottom row), or simultaneously in both. The results are illustrated in Fig. 4, which reports the mean simple regret as a function of normalised sampling cost, aggregated over 20 independent runs.

Specifically, our results suggest that MSBO's advantage is not confined to specific complexity patterns: the ability to decouple upstream and downstream exploration, and to commit second-stage resources only when justified by intermediate observations, yields systematic improvements. This is evident not only when the initial process ( $f_1$ ) is more complex (top-left corner), but crucially, even when the second process ( $f_2$ ) is the most difficult to regress (bottom-right corner). This observation suggests substantial opportunities and potential for applying MSBO in scenarios where no highly informative proxy measurement is available. Even in such cases, low-fidelity and





**Fig. 3** MSBO achieves superior data efficiency by selectively allocating samples to promising regions of the parameter space. (A) Aggregated log-regret over 10 independent runs as a function of sampling cost (uniform cost model), with shaded regions indicating a single standard deviation interval on both sides of the mean. Both cost and regret are unitless. (B) and (C) show a representative sampling distribution for standard BO (B) and MSBO (C), respectively. The left subplots display the first-stage intermediate outputs  $h_1$  (dots) plotted against the input  $x_1$ , overlaid on the true first-stage function (blue line). The right subplot shows the locations of the second-stage evaluations within the  $h_1 \times x_2$  domain. Markers in both plots are colour-coded by the sample ID (iteration number), with black dots representing the samples of the random initialisation.

not directly related observations, additional experiments, as well as simulations, can improve the overall sample efficiency of complex and costly experiments of the actual objective function.

Furthermore, we observe that the relative sampling frequency of first-stage and second-stage evaluations adjusts dynamically, demonstrating the framework's ability to allocate

resources depending on the relative complexity of the stages. We report the number of samples drawn from each of the two subprocesses for each setting in SI B, Fig. S3.

**4.2.3 Impact of heterogeneous costs.** A common scenario in experimental campaigns is that downstream characterisation, such as device fabrication, purification, or final testing,



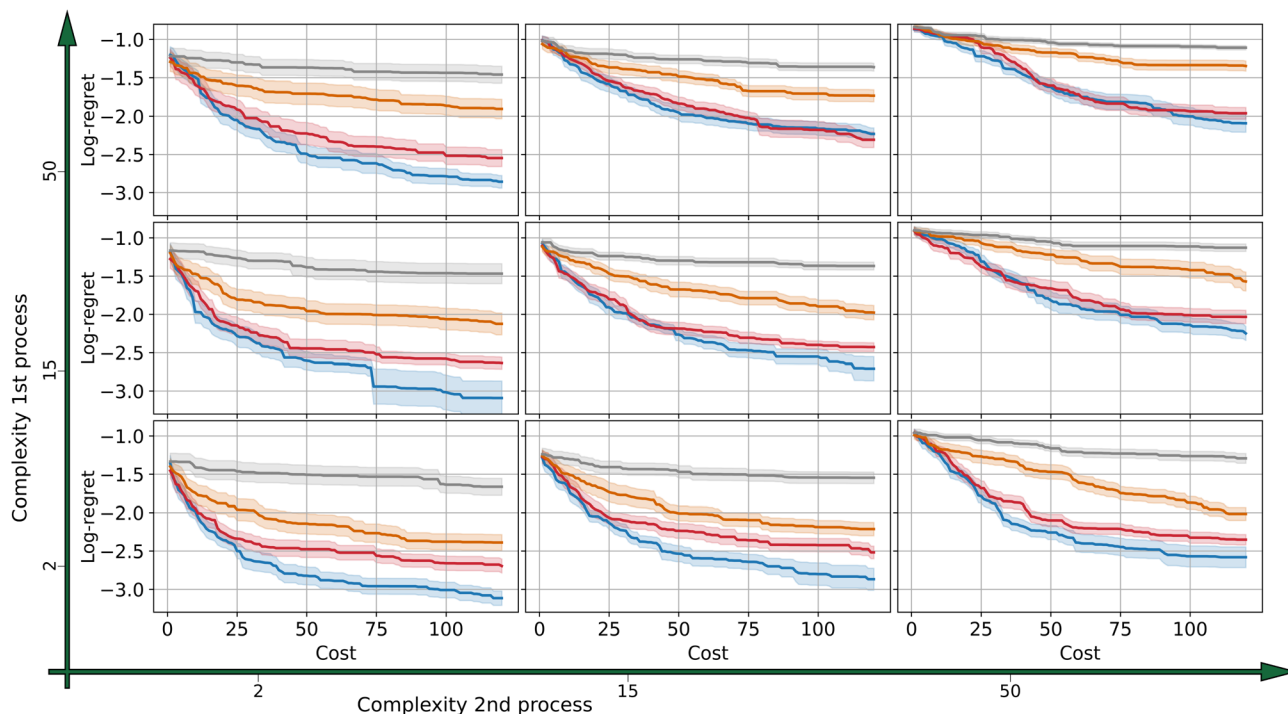


Fig. 4 Log-regret plotted against normalised sampling cost (uniform cost model) for a two-stage cascade process. All results are aggregated over 20 optimisation runs. Shaded regions indicate a 1 standard deviation interval. The rows correspond to increasing complexity levels for the first stage ( $\mathcal{Q}_{\text{seed}}$  sizes 2, 15, 50). The columns correspond to increasing complexity levels for the second stage ( $\mathcal{Q}_{\text{seed}}$  sizes 2, 15, 50). Each panel contains performance curves for MSBO (blue), standard BO (orange), BOFN (red), and random search (grey). Cost and regret are considered unitless here.

is substantially more expensive than upstream preparation or proxy stages, which often involve inexpensive computations or early-stage synthesis.

To evaluate how robust MSBO remains under such realistic conditions, we examine its behaviour when the relative costs of the two stages are varied. We focus on the process defined by complex upstream dynamics (corresponding to the top-left case in Fig. 4) and compare three regimes: a uniform-cost setting (ratio 1 : 1); a downstream-heavy heterogeneous-cost setting (ratio 1 : 10), where the second stage is ten times more expensive than the first; and an upstream-heavy setting, representing the inverse scenario (ratio 10 : 1) (see Fig. 5). As detailed in Section 4.1, the total cost of a full cascade evaluation is normalised to 1 in all experiments. Consequently, the baseline methods (standard BO and BOFN), which do not support dynamic early stopping, exhibit identical behaviour regardless of the internal cost distribution. By contrast, MSBO naturally adapts to the cost asymmetry through its resumable sampling mechanism. Even though MSBO does not explicitly incorporate cost parameters into the acquisition function, its resumable sampling mechanism allows it to implicitly capitalise on low-cost intermediate evaluations.

Empirically, MSBO exhibits improved data efficiency in the heterogeneous-cost regime compared to the uniform one. The MSBO algorithm relies more heavily on the inexpensive first stage to discard unpromising candidates early, and only progresses samples to the costly second stage when their intermediate results indicate sufficiently high utility. This behaviour

leads to faster convergence toward better-performing solutions. Furthermore, in the inverted 10 : 1 cost regime, while the absolute budget saved by early stopping inherently diminishes

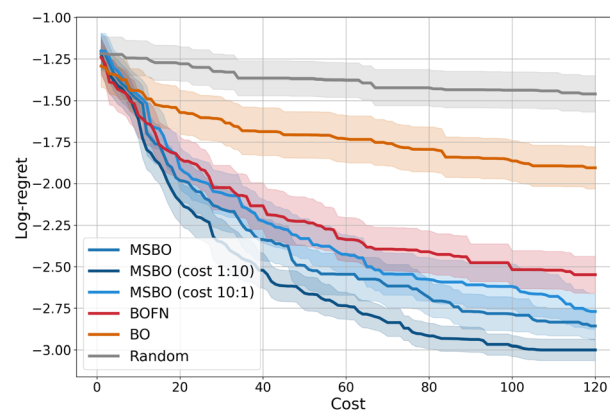


Fig. 5 Log-regret versus cumulative cost for a two-stage process with first- and second-stage complexities of 50 and 2, respectively. The light blue curve represents MSBO in the upstream-heavy heterogeneous cost setting, where the first stage is ten times more expensive than the second (ratio 10 : 1); the intermediate blue curve, the standard uniform cost setting (ratio 1 : 1); and the dark blue curve, the downstream-heavy setting (ratio 1 : 10). The baselines always execute the full cascade sequence; as the total cost is normalized to 1, their trajectories are identical in all settings. The trajectories represent the aggregated results over 20 optimisation runs. Shaded regions indicate a 1 standard deviation interval. Cost and regret are considered unitless here.



due to the low cost of the downstream stage, MSBO retains a strict performance advantage over both standard BO and BOFN. This demonstrates that the flexibility gained by decoupling the sampling processes allows MSBO to systematically outperform baselines irrespective of whether the first or second stage constitutes the primary cost driver.

**4.2.4 Three-stage experiments and process noise.** We conduct further experiments to assess the framework's scalability to longer cascades and its robustness in more realistic settings, specifically addressing process noise (imperfectly reproducible experiments) and partial observability of sample states ( $\mathbf{M} \neq \mathbf{I}$ ). MSBO inherently relies upon the fidelity of intermediate measurements. Propagating measurement noise ( $\epsilon^m$ ) at intermediate stages through the surrogate cascade could, in principle, become counterproductive if disproportionately higher than the noise at the objective stage. Conversely, we expect the algorithm to remain structurally robust against inherent process noise ( $\epsilon^p$ ), as the capacity to sample high-variance upstream stages independently facilitates the cost-efficient modelling of the underlying distributions.

When evaluating our model on a three-stage cascade process, we observe that the performance advantages of MSBO, as demonstrated in the two-stage setting, extend effectively to this deeper process, where it consistently outperforms the baselines (see aggregated results in Fig. S4). To test the impact of partial observability of intermediate results, we evaluate our model in a more complex three-stage cascade, where each intermediate process outputs a four-dimensional state, of which only the first two dimensions are accessible to the optimiser. Consequently, half of the information required to fully model each stage is missing. Due to the increased complexity and the information loss, both standard BO and BOFN degrade to the performance level of random search, whereas MSBO remains significantly more efficient (see Fig. S5). This result is significant, as it reflects the more realistic experimental setting, where not the entire intermediate result of an experiment is observable, but only aspects of it, *e.g.* a thin film material is generated, but only specific aspects of it are observed (*e.g.* conductivity or absorption), but not all details (microstructure, interfaces, other thin-film properties, *etc.*).

Finally, to assess robustness in realistic laboratory settings, we evaluate performance on a two-stage cascade with process noise  $\epsilon_1^p$ . MSBO demonstrates superior robustness in this challenging regime compared to the baselines, consistently identifying high-performing solutions by effectively leveraging intermediate measurements and flexibly sampling from the various subprocesses. Furthermore, calculating the denoised performance of the sample with the best observed objective  $y$  reveals that relying on single observations often leads to selecting suboptimal parameters driven by favourable noise outliers. Conversely, selecting the final parameters based on the surrogate model's maximum predicted mean among the collected samples consistently yields superior performance. This result suggests that in noisy real-world environments, a model-derived policy for selecting the final best-performing set of parameters is significantly more reliable than one based strictly on historical observations, as the model effectively filters

out stochastic fluctuations. This advantage is particularly pronounced for MSBO. Because MSBO models the underlying process structure more accurately than the baselines, the performance gain achieved by relying on its surrogate, rather than on single stochastic outcomes, is significantly larger (see Fig. S6).

### 4.3 Real-world applications

We evaluate MSBO on three datasets representative of chemical optimisation tasks, comprising both computational and experimental workflows. These case studies exemplify the relevance of our method for real-world discovery campaigns, where multi-stage processes enable cost-efficient navigation of, for example, high-dimensional chemical spaces by bridging the gap between lower-fidelity proxies and expensive validations. In all scenarios, the objective is the optimisation of specific molecular properties. To manage the high dimensionality of the chemical space, we generate Morgan fingerprints (1024 dimensions, depth 3) for all molecules using RDKit<sup>40</sup> and compress these representations to the 16 most informative principal components using PCA. Furthermore, to prevent the undersampling of the objective stage in scenarios where a relatively simple downstream process is dominated by high uncertainty propagating from complex upstream stages, we enforce a minimum stage-sampling frequency constraint across all real-world tasks; specifically, we ensure the objective function is evaluated at least once for every five queries of the preceding stage.

**4.3.1 QM9 (HOMO and LUMO levels).** We target the minimisation of the energy levels of the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO) using the QM9 dataset.<sup>41</sup> These energy levels are critical indicators of optoelectronic behaviour and chemical reactivity. The property values in the dataset are derived from a high-fidelity, two-stage computational approach.<sup>42,43</sup> This task naturally aligns with a multi-stage workflow: the initial stage employs a density functional theory calculation using the Perdew–Burke–Ernzerhof exchange–correlation functional with an aug-cc-TZVP basis set. This acts as a lower-fidelity stage-1 observation required for the second, high-fidelity stage, which involves refining the result using an accurate, but computationally expensive, eigenvalue-self-consistent GW approach (ev-GW@Perdew–Burke–Ernzerhof/(aug-cc-DZVP  $\rightarrow$  aug-cc-TZVP)).<sup>44</sup> We estimate the computational cost of the initial DFT simulation to be approximately 1/10 of the expensive GW correction.

**4.3.2 AqSolDB (solubility).** For the minimisation of aqueous solubility, we utilise the AqSolDB dataset,<sup>45</sup> a large-scale, high-quality compilation focused on aqueous solubility values ( $\log S$ ) for nearly 10 000 unique chemical compounds. The database aggregates and curates data from nine different public datasets and includes corresponding molecular representations. In our optimisation setup, we employ the partition coefficient ( $\log P$ ), computed *via* RDKit, as a proxy for the experimental solubility values. Given that the computational cost of calculating  $\log P$  is negligible compared to the cost of



performing wet-lab experiments, we treat the first-stage cost as effectively zero. We note that this specific experiment serves as a conceptual demonstration using real-world data. In scenarios where a proxy incurs negligible computational cost, and the design space is finite and discrete, exhaustive proxy evaluation remains preferable to any adaptive strategy. However, MSBO is explicitly designed for typical SDL conditions where exhaustive evaluation is not possible; such constraints arise either because the design space is continuous (*e.g.*, flow rates, temperatures), rendering exhaustive enumeration impossible, or because the proxy itself incurs a non-trivial computational or physical cost. In this setting, we use the residual connection configuration for our surrogate model (as described in Section 3.1) to mitigate the information loss inherent in compressing the molecular features into a single scalar proxy.

**4.3.3 FreeSolv (hydration energy).** We minimise the hydration free energy using the FreeSolv database,<sup>46</sup> which contains values for the aqueous free energy of solvation ( $\Delta G_{\text{solv}}$ ) for small, drug-like molecules. This property quantifies the change in free energy when a molecule moves from the gas phase to an aqueous solution. The database compiles two types

of data: experimental values derived from a literature survey of reliable, peer-reviewed sources, and calculated values derived from alchemical free energy calculations performed using molecular dynamics simulations. The computational values were generated using the GAFF small molecule force field in TIP3P water with AM1-BCC charges, with simulations executed using the GROMACS package. In our experiments, the objective is to optimise the experimental value, using the values derived from MD simulations as the intermediate proxy. We estimate the cost of the simulation to be 1/50 of the cost of the physical experiment.

**4.3.4 Results.** MSBO consistently outperforms the baselines, identifying molecules within the top 1% of the dataset significantly faster. We quantify this performance across all four optimisation tasks by tracking the percentiles of the best-found solutions relative to the total dataset size as a function of accumulated cost (see Fig. 6). Notably, for all tasks excluding FreeSolv, MSBO discovers molecules in the top 1% using only half of the budget required by standard BO. Moreover, the advantage in cost grows larger for lower percentiles, and MSBO is the only method that consistently identifies samples within

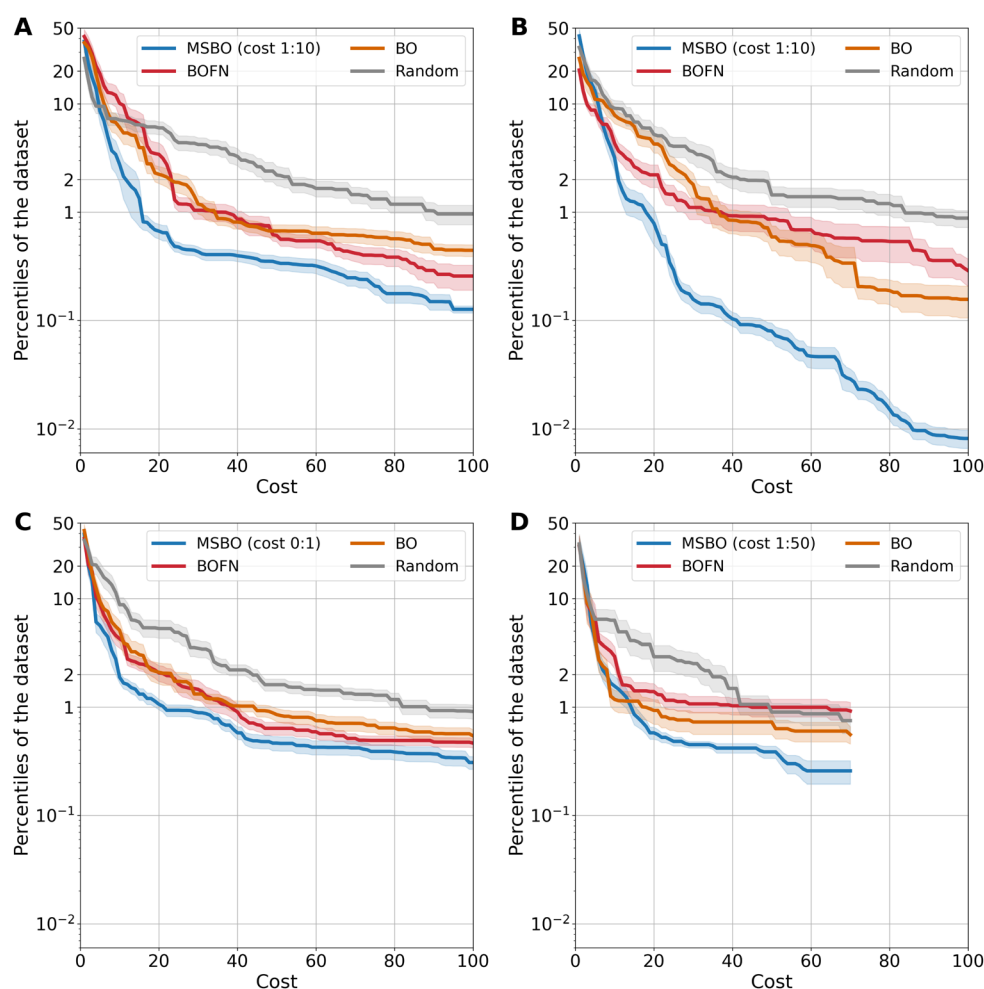


Fig. 6 Percentile of the best-found molecule relative to the total dataset size as a function of accumulated cost. All results are aggregated over 30 optimisation runs. Shaded regions indicate a 1 standard deviation interval. The properties optimised, and the relative datasets are: (A) HOMO energy levels (QM9); (B) LUMO energy levels (QM9); (C) aqueous solubility (AqSolDB); and (D) hydration free energy (FreeSolv).



the top 0.3% in all datasets, within the allocated budget. This advantage is most dramatic in the specific case of the LUMO property, where MSBO navigates to the top 0.01% of candidates, an order of magnitude lower than standard BO (see Panel B).

Furthermore, the comparison with BOFN reveals that simply modelling the cascade structure is insufficient. In the FreeSolv task, BOFN performs worse than standard BO, whereas MSBO's resumable sampling strategy effectively leverages the proxy measurements to outperform the baselines (see Panel D). Finally, MSBO exhibits the most rapid optimisation trajectory in the early stages (costs 0–20), indicating superior initial exploration and immediate identification of high-utility regions. These results confirm that the cascade structure provides a robust navigational signal, enabling efficient optimisation even with high-dimensional molecular representations and varying regimes of proxy fidelity and cost.

## 5 Conclusions

In this work, we introduced MSBO, a Bayesian optimisation framework designed to enhance dynamic decision-making in self-driving laboratories (or multi-step computational workflows) with multi-stage cascade processes. Unlike standard Bayesian optimisation, which treats experimental workflows as rigid, monolithic black boxes, MSBO explicitly models the structure of cascade processes. By integrating a cascade GP surrogate with a nested acquisition function and an inventory management system, our approach enables resumable sampling, allowing the optimiser to selectively advance promising candidates while discarding uninformative ones at early stages.

We validated the framework through extensive synthetic benchmarking, demonstrating that MSBO consistently achieves lower regret than standard baselines and network-based alternatives (BOFN), and using a smaller budget. These experiments confirmed the method's robustness across various regimes of stage complexity and heterogeneous cost ratios. Furthermore, our evaluation of real-world optimisation tasks, including the optimisation of molecular solubility and electronic properties, revealed significant efficiency gains. Notably, MSBO identified top-performing molecules with considerably fewer resources than traditional methods, validating its practical utility in high-dimensional search spaces.

Ultimately, these results suggest that incorporating intermediate proxy measurements is a critical step toward more autonomous and efficient materials discovery. By bridging the gap between cheap computational approximations and expensive experimental validations, MSBO offers a scalable pathway for hybrid computational-experimental self-driving labs. The MSBO approach can replace (potentially biased) multi-stage funnel approaches, where selection criteria need to be defined beforehand, and can thus efficiently learn to optimise in high-dimensional parameter spaces with multiple intermediate observations. This opens possibilities to extend self-driving labs from confined optimisation tasks to more complex open challenges of autonomous discovery.

## Author contributions

L. T. developed the methodology, implemented the software, and conducted the computational experiments (conceptualisation, methodology, software, investigation, data curation, visualisation, writing – original draft). P. F. supervised the project, secured funding, and provided critical revisions to the manuscript (conceptualisation, methodology, supervision, resources, funding acquisition, writing – review & editing). Both authors contributed to the study design and preparation of the final work.

## Conflicts of interest

There are no conflicts to declare.

## Data availability

The code for the MSBO method, including scripts to recreate the synthetic datasets used in this study, is available on [GitHub](#) at [aimat-lab/Multi-stage-Bayesian-optimisation](#). The version of the code used to generate the results in this paper has been archived on Zenodo with <https://doi.org/10.5281/zenodo.18926239> (v1.0.0).

Supplementary information (SI) is available. See DOI: <https://doi.org/10.1039/d5dd00572h>.

## Acknowledgements

We acknowledge support by the Federal Ministry of Education and Research (BMBF) under Grant No. 01DM21002A (FLAIM). We acknowledge support by the Federal Ministry of Education and Research (BMBF) under Grant No. 01DM21001B (German-Canadian Materials Acceleration Center). We thank Henrik Schopmans for his helpful comments on the manuscript. This work was performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

## Notes and references

- 1 H. Kitano, *npj Syst. Biol. Appl.*, 2021, 7, 29.
- 2 M. L. Green, I. Takeuchi and J. R. Hattrick-Simpers, *J. Appl. Phys.*, 2013, 113, 231101.
- 3 D. C. Montgomery, *Design and analysis of experiments*, John Wiley & sons, 2017.
- 4 M. Abolhasani and E. Kumacheva, *Nat. Synth.*, 2023, 1–10.
- 5 P. M. Maffettone, P. Friederich, S. G. Baird, B. Blaiszik, K. A. Brown, S. I. Campbell, O. A. Cohen, T. Collins, R. L. Davis and I. T. Foster, *Digital Discovery*, 2023, 2, 1644–1659.
- 6 F. Delgado-Licona and M. Abolhasani, *Adv. Intell. Syst.*, 2023, 5, 2200331.
- 7 D. Salley, G. Keenan, J. Grizou, A. Sharma, S. Martín and L. Cronin, *Nat. Commun.*, 2020, 11, 2771.



- 8 P. Rajak, A. Krishnamoorthy, A. Mishra, R. Kalia, A. Nakano and P. Vashishta, *npj Comput. Mater.*, 2021, 7, 108.
- 9 P. M. Maffettone, J. K. Lynch, T. A. Caswell, C. E. Cook, S. I. Campbell and D. Olds, *Mach. Learn.: Sci. Technol.*, 2021, 2, 025025.
- 10 A. A. Volk, R. W. Epps, D. T. Yonemoto, B. S. Masters, F. N. Castellano, K. G. Reyes and M. Abolhasani, *Nat. Commun.*, 2023, 14, 1403.
- 11 B. P. MacLeod, F. G. Parlane, T. D. Morrissey, F. Häse, L. M. Roch, K. E. Dettelbach, R. Moreira, L. P. Yunker, M. B. Rooney, J. R. Deeth, *et al.*, *Sci. Adv.*, 2020, 6, eaaz8867.
- 12 A. E. Gongora, B. Xu, W. Perry, C. Okoye, P. Riley, K. G. Reyes, E. F. Morgan and K. A. Brown, *Adv. Sci.*, 2020, 6, eaaz1708.
- 13 B. J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J. I. M. Alvarado, J. M. Janey, R. P. Adams and A. G. Doyle, *Nature*, 2021, 590, 89–96.
- 14 B. P. MacLeod, F. G. Parlane, C. C. Rupnow, K. E. Dettelbach, M. S. Elliott, T. D. Morrissey, T. H. Haley, O. Proskurin, M. B. Rooney, N. Taherimakhosousi, *et al.*, *Nat. Commun.*, 2022, 13, 995.
- 15 A. Agresti, F. Di Giacomo, S. Pescetelli and A. Di Carlo, *Nano Energy*, 2024, 122, 109317.
- 16 J. Wu, L. Torresi, M. Hu, P. Reiser, J. Zhang, J. S. Rocha-Ortiz, L. Wang, Z. Xie, K. Zhang, B.-w. Park, *et al.*, *Science*, 2024, 386, 1256–1264.
- 17 R. Lam, D. L. Allaire and K. E. Willcox, *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015, p. 0143.
- 18 J. Song, Y. Chen and Y. Yue, *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 3158–3167.
- 19 M. Poloczek, J. Wang and P. Frazier, *Adv. Neural Inf. Process. Syst.*, 2017, 30, 4288–4298.
- 20 M. C. Kennedy and A. O'Hagan, *Biometrika*, 2000, 87, 1–13.
- 21 A. I. Forrester, A. Söbester and A. J. Keane, *Proceedings of The Royal Society a: Mathematical, Physical and Engineering Sciences*, 2007, vol. 463, pp. 3251–3269.
- 22 K. Kandasamy, G. Dasarathy, J. Oliva, J. Schneider and B. Poczos, *J. Artif. Intell. Res.*, 2019, 66, 151–196.
- 23 K. Kandasamy, G. Dasarathy, J. Schneider and B. Poczos, *International conference on machine learning*, 2017, pp. 1799–1808.
- 24 R. Astudillo and P. Frazier, *International Conference on Machine Learning*, 2019, pp. 354–363.
- 25 R. Astudillo and P. Frazier, *Adv. Neural Inf. Process. Syst.*, 2021, 34, 14463–14475.
- 26 P. Buathong, J. Wan, R. Astudillo, S. Daulton, M. Balandat and P. I. Frazier, *International Conference on Machine Learning*, 2024, pp. 4752–4784.
- 27 P. Buathong and P. I. Frazier, *arXiv*, preprint, arXiv:2506.11456, 2025, DOI: [10.48550/arXiv:2506.11456](https://doi.org/10.48550/arXiv:2506.11456).
- 28 T. D. Nguyen, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, K. J. Deane and P. G. Sanders, *Australasian Joint Conference on Artificial Intelligence*, 2016, pp. 268–280.
- 29 S. Kusakawa, S. Takeno, Y. Inatsu, K. Kutsukake, S. Iwazaki, T. Nakano, T. Ujihara, M. Karasuyama and I. Takeuchi, *Neural Comput.*, 2022, 34, 2408–2431.
- 30 L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh and A. Talwalkar, *J. Mach. Learn. Res.*, 2018, 18, 1–52.
- 31 S. Falkner, A. Klein and F. Hutter, *International conference on machine learning*, 2018, pp. 1437–1446.
- 32 K. Swersky, J. Snoek and R. P. Adams, *arXiv*, preprint, arXiv:1406.3896, 2014, DOI: [10.48550/arXiv.1406.3896](https://doi.org/10.48550/arXiv.1406.3896).
- 33 E. Brochu, V. M. Cora and N. De Freitas, *arXiv*, preprint, arXiv:1012.2599, 2010, DOI: [10.48550/arXiv.1012.2599](https://doi.org/10.48550/arXiv.1012.2599).
- 34 S. W. Ober, C. E. Rasmussen and M. van der Wilk, *Uncertainty in Artificial Intelligence*, 2021, pp. 1206–1216.
- 35 C. E. Rasmussen, C. K. Williams *et al.*, *Gaussian processes for machine learning*, Springer, 2006, vol. 1.
- 36 H. Liu, Y.-S. Ong, X. Shen and J. Cai, *IEEE Trans. Neural Netw. Learn. Syst.*, 2020, 31, 4405–4423.
- 37 E. V. Bonilla, K. Chai and C. Williams, *Adv. Neural Inf. Process. Syst.*, 2007, 20, 153–160.
- 38 M. A. Alvarez, L. Rosasco, N. D. Lawrence, *et al.*, *Trends Mach. Learn.*, 2012, 4, 195–266.
- 39 N. Srinivas, A. Krause, S. M. Kakade and M. Seeger, Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design, *ICML 2010*, 2010, pp. 1015–1022.
- 40 G. Landrum, *et al.*, *RDKit*, 2024, DOI: [10.5281/zenodo.10633624](https://doi.org/10.5281/zenodo.10633624).
- 41 R. Ramakrishnan, P. O. Dral, M. Rupp and O. A. Von Lilienfeld, *Sci. Data*, 2014, 1, 1–7.
- 42 A. Fediai, P. Reiser, J. E. O. Peña, P. Friederich and W. Wenzel, *Sci. Data*, 2023, 10, 581.
- 43 A. Fediai, P. Reiser, J. E. O. Peña, W. Wenzel and P. Friederich, *Mach. Learn.: Sci. Technol.*, 2023, 4, 035045.
- 44 L. Ruddigkeit, R. Van Deursen, L. C. Blum and J.-L. Reymond, *J. Chem. Inf. Model.*, 2012, 52, 2864–2875.
- 45 M. C. Sorkun, A. Khetan and S. Er, *Sci. Data*, 2019, 6, 143.
- 46 D. L. Mobley and J. P. Guthrie, *J. Comput.-Aided Mol. Des.*, 2014, 28, 711–720.

