Applying Multi-Fidelity Bayesian Optimization in Chemistry: Open Challenges and Major Considerations

Edmund Judge
Department of Chemistry
Imperial College London
United Kingdom
ewj23@ic.ac.uk

Mohammed Azzouzi
Computational Molecular Design Laboratory
EPFL
Switzerland
mohammed.azzouzi@epfl.ch

Austin M. Mroz
Department of Chemistry
Imperial College London
United Kingdom
a.mroz@ic.ac.uk

Antonio Del Rio Chanona
Department of Chemical Engineering
Imperial College London
United Kingdom
adelrioc@ic.ac.uk

Kim E. Jelfs
Department of Chemistry
Imperial College London
United Kingdom
k.jelfs@ic.ac.uk

Abstract

Multi-fidelity Bayesian optimization (MFBO) leverages experimental and/or computational data of varying quality and resource cost to optimize towards desired maxima cost-effectively. This approach is particularly attractive for chemical discovery due to MFBO's ability to integrate diverse data sources. Here, we investigate the application of MFBO to accelerate the identification of promising molecules or materials for target applications. We specifically analyze the conditions under which lower-fidelity data can enhance performance compared to single-fidelity problem formulations through four case studies – addressing two key challenges: i) understanding the impact of the correlation between cost and data fidelity, ii) assessing the impact of the acquisition function.

1 Introduction

From new drug molecules, to novel battery technologies, the discovery of new molecules and materials is critical to addressing the global challenges that humanity faces today. Yet, the chemical space that we face is massive; for example, it is estimated that there are 10^{60} potential small molecules that could be synthesised. This excludes the number of potential materials that could be made from these molecules, and the optimal experimental conditions for each. This chemical space is far too vast to only search with chemical intuition and the conventional trial-and-improvement process, particularly given laboratory experiments are typically slow (up to months) and costly. Efficiently traversing chemical space can often be translated into an optimization problem, where we optimize towards improved performance, reaction yield, novelty, *etc*. Different computational and experimental tools can be used to evaluate the performance of candidate chemical systems, and optimization algorithms then used to select the next best candidates to test. Derivative-free, surrogate-based optimization algorithms are of particular relevance to this task – specifically, Bayesian optimization (BO).

BO is a resource-aware optimization technique that balances exploration of the parameter space (sampling sparse data regions) with exploitation (sampling regions rich in data) [1; 2]. BO has enjoyed a recent resurgence due to advances in computational power that make its implementation more accessible. In practice, implementation is accomplished by using surrogate models that encode

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

a distribution of predicted values, rather than point estimates, usually a Gaussian Process (GP). With this information, BO will balance exploration and exploitation, and compute the next candidate to evaluate in the search space. Within chemistry, BO has been applied to a wide variety of problems, including for antimicrobial polymer design [3], electrolyte optimization in zinc-ion batteries [4; 5], and nanoporous materials, [6; 7] among others [8; 9].

While powerful, there are several important considerations when applying BO for molecular discovery. First, the datasets used in this domain are often high-dimensional, which causes the surrogate model to suffer from the "curse of high-dimensionality"[10]. Specifically, the volume of the search space grows exponentially with the number of dimensions, and therefore the model has to encounter greater uncertainty, which adversely impacts its accuracy. Second, molecular components must be encoded in a machine-readable format. There are several options available, such as SMILES [11], SELFIES [12], molecular graphs, [13] and computer-learned representations, [14; 15] but each has its trade-offs. For example, SMILES are a succinct representation of a molecule using ACSII characters, and are therefore cheap to store and amenable to algorithms from the field of natural language processing; however, critical details of the 3-dimensional configuration of the molecule are lost in this 2-dimensional representation, nor is any feature information communicated - the model having to infer this instead. Furthermore, SMILES strings are non-canonical: this means a molecule can have two distinct SMILES representations that are more varied than two SMILES for totally different molecules[14]. More geometric and precise representations, such as a chemical table, where the coordinates of each atom in the molecule are recorded in 3-dimensions, also have their drawbacks, since the molecule's invariance to rotations is not intrinsically captured in the representation. Such inconsistencies can easily confuse a model. Third, chemical design problems often feature mixed-domains [16; 17] (*i.e.* domains containing both categorical and discrete features), which require additional considerations for BO problem formulation.

Researchers often have access to several different experiment types that may render related information at different costs[18]. The plethora of computational techniques already available, of various costs and accuracy, invites the possibility of using the information provided (typically property predictions) to supplement the more expensive laboratory experiments. This idea is captured in the notion of a data's fidelity, *i.e.* the accuracy of the data relative to the true quantity. Typically, high-fidelity data is more expensive to obtain, as the cost frequently correlates with the accuracy. Therefore, here we consider multi-fidelity BO (MFBO) and examine open challenges associated with the technique [19; 20; 21; 22; 23]. Specifically, we examine the circumstances that lead to MFBO outperforming single-fidelity BO (SFBO). Two challenges are considered, i) the selection of the acquisition function, and ii) the cost of the low-fidelity data and its correlation to the high-fidelity data.

2 Experimental Methods

The major difference between the single- and multi-fidelity case is that MFBO enables experiments of varying cost to be performed. Consequently, the aim is no longer to reach the optimum in the fewest number of iterations, but rather to do so while exhausting the smallest budget. Lower-fidelity evaluations usually incur a lower cost, so provided there is some correlation between the low-fidelity and the high-fidelity target, it is possible to reach the optimum more economically (*i.e.* using fewer resources). Here, we briefly outline the general algorithmic framework of MFBO; specifically, we describe the surrogate model and acquisition function(s) that we studied. We encourage readers to engage with more detailed descriptions in the literature for more information[7; 19; 20].

Any model can be used as the surrogate model, providing it encodes some level of uncertainty, which is critical for later stages of the BO algorithm. Although alternatives do exist, such as Bayesian neural networks [24; 25], here we focused on using GPs as the surrogate model [26]. Specifically we use the implementation of the multi-fidelity GP with down-sampling in BoTorch (*SingleTaskMultiFidelityGP*) 27; 28 . The choice of the GP kernel is important to the optimization performance; Appendix A.2 includes a detailed description of the SFBO and MFBO algorithm frameworks.

We explore three acquisition functions: i) Multi-Fidelity Maximum Entropy Search (MF-MES) [29], which measures the gain in mutual information between the candidate element and the maximum function value, ii) Multi-Fidelity Targeted Variance Reduction (MF-TVR) [19], which suggests the element and fidelity that minimize the variance of the model's prediction at the point with the greatest

expected improvement after sufficient scaling, and iii) Multi-Fidelity Custom (MF-Custom), a custom acquisition function which normalizes and then combines the outputs of the above two techniques, i.e.

$$\textit{MF-Custom}(X) = \left(\frac{\textit{MF-MES}(X)}{||\textit{MF-MES}(X)||} + \frac{\textit{MF-TVR}(X)}{||\textit{MF-TVR}(X)||} \right), \tag{1}$$

where $||\cdot||$ is the Euclidean-norm and $X = [\vec{x_1}, \ldots, \vec{x_n}]^T$ is an array of elements, $\vec{x_i}$, from the search-space. We did not include the Knowledge Gradient [1] acquisition function here, as preliminary tests showed the computational cost of the approach was prohibitive for our purposes in molecular discovery. Finally, we compared MFBO to SFBO by running Single-Fidelity Expected-Improvement (SF-EI) alongside each of the above multi-fidelity acquisition functions (see Garnett (2023) for a definition).

3 Results and Discussion

3.1 Impact of Problem Formulation on Performance

We first compare the performance of the different MFBO algorithms with SFBO across four different problems to identify the impact of problem formulation on MFBO performance. Here we consider two synthetic problems and two datasets relevant to chemical discovery. The first two synthetic problems, are relevant to cases where the design space is continuous such as reaction conditions [8] and thin film deposition conditions [30].

Problem 1: The first relatively simple problem, involves the synthetic RKHS function [31], with the domain, $[0, 1]$, divided into 500 evenly distanced points and low-fidelity data generated by adding Gaussian noise to the high-fidelity evaluations. The function is good for benchmarking BO as it has multiple local maxima to challenge the optimization. We gave an assigned cost of 0.1 to the low-fidelity data relative to the high-fidelity data and there was a correlation of 0.88 between the two fidelities.

Problem 2: The second synthetic problem is more challenging, as we consider the 6D negated Hartmann function [32]. Like before, this is a good benchmarking function, as it has multiple local maxima, as well as the new difficulty of additional dimensions. We produced a dataset using the 6D negated Hartmann function as defined by the BoTorch library [27], with the lower-fidelity data created by adding Gaussian noise to the exact, high-fidelity evaluations (with correlation 0.76) and an assigned relative cost of 0.1.

Problem 3: For the first chemical discovery problem, we used the dataset employed by Gantzler *et al.* when applying MFBO to locate covalent organic frameworks (COFs), a class of porous materials, with the largest equilibrium absorptive selectivity for xenon and krypton at room temperature (see Gantzler *et al.* (2023) for more details). The lower-fidelity data was created using Henry's law (as opposed to Markov chain Monte Carlo simulation in the binary grand-canonical ensemble as used for the high-fidelity case) and found to have a correlation 0.97 to the higher-fidelity data, and we assigned a relative cost of 0.2. This cost is higher to what is defined in the original paper as we were trying to measure the performance of our custom acquisition function against that defined by Gantzler *et al.*, and therefore created tougher conditions for a more conservative comparison. Each molecule was represented by 4 structural features and 10 compositional features, giving a 14 dimensional vector representation.

Problem 4: Finally, we looked at a problem using a dataset of organic molecules for organic photovoltaic applications. We built a database of 50,000 molecules using our *stk* supramolecular toolkit software [33] that assembles such systems from building blocks. We included common building blocks used in the organic photovoltaic community. The high-fidelity data was property data for these systems computed using the extended-tight binding (xtb) technique [34], and the lower-fidelity data was the product of a machine-learning model with correlation 0.91, which we assigned a relative cost of 0.1. We represent the molecules as a 72-dimension array of concatenated arrays of building block descriptors calculated using xtb. See Figures 4 - 7 in the appendix for more details of the various datasets, and for a deeper discussion on `STK_search`.

Problems 1 and 2 were seeded with 5 initial samples (both high- and low-fidelity evaluations), afforded a budget of 50, and had a domain-size of 500. Problem 3 was seeded with 3 initial samples

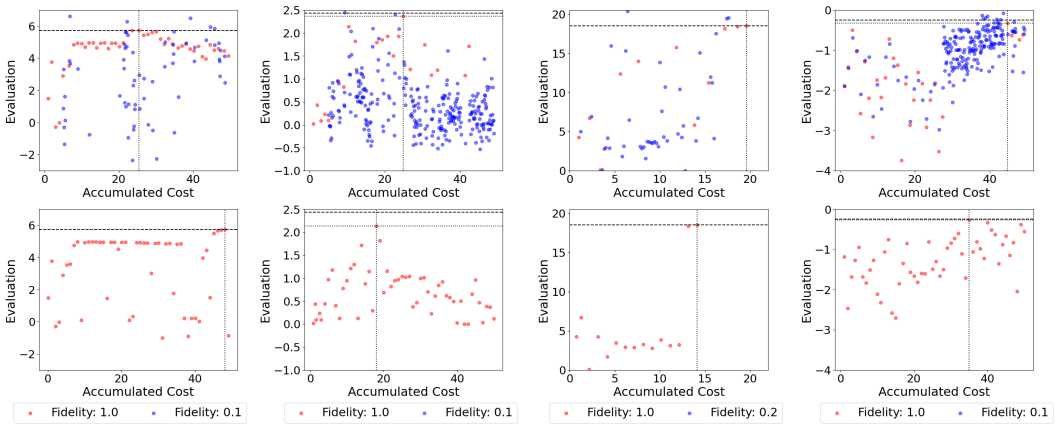


Figure 1: The behaviour of the MF-MES (top) and SF-EI (bottom) search-algorithms for a single run optimizing (far-left) Problem 1 (RKHS), (middle-left) Problem 2 (6D negated Hartmann), (middle-right) Problem 3 (COF selectivities) and (far-right) Problem 4 (organic photovoltaic molecules). The different dashed lines denote the domain optimum and the obtained optimum.

and allowed to run until the optimum was obtained and had a domain-size of 608. Problem 4 was seeded with 25 initial samples, afforded a budget of 50 and had domain-size of 44928.

Figure 1 illustrates the results of a single run for each of the problems using the MF-MES and SF-EI acquisition functions. MF-MES does not consistently outperform SF-EI, which is surprising as one would expect the presence of the lower-fidelity data to be beneficial. In Problems 1 and 4, there is an improvement in performance with MF-MES obtaining the optimum after a budget of only 25.4 is exhausted (compared to 48.1 for SF-EI), and 9.8 (compared to 35 for SF-EI), respectively. For Problem 4, the domain optimum is not attained in any search. In Problems 2 and 3, MF-MES does not outperform SF-EI, and therefore the presence of the lower-fidelity information does not benefit the optimization process.

3.2 Fidelity Correlation and Cost

As shown above, the performance of MFBO depends on the dataset we tackle. The different datasets have different correlations between the high- and low-fidelity data, as well as different problem complexities. To first assess the impact of the cost and the data fidelity correlation, we varied these parameters in the two synthetic problems to see how it affected the relative MFBO performance compared to a single-fidelity search. To do this, new sample-spaces were created for Problems 1 and 2, featuring different quantities of Gaussian noise to alter data correlation. Figure 2 reveals the benefits of high-correlation and low-cost for the MF-MES search-algorithm. Clearly, if the low-fidelity data is highly correlated with the objective function, then the search-algorithm benefits from access to this cheap information. However, what is surprising is how high the correlation and how low the cost of the lower-fidelity data has to be to give the algorithm any sort of advantage over SFBO. Indeed, for Problems 1 and 2, we see only 4 and 3 instances, respectively, where the Relative Improvement is less than 1. We define Relative Improvement as the budget exhausted before obtaining the optimum (averaged over 5 runs, with an allocated budget of 50, and assigned a score of 60 if the optimum was not reached) divided by the budget exhausted by the SF-EI acquisition function (48.1 and 12.7 for Problems 1 and 2, respectively). This is a result seldom discussed explicitly in the literature, but does explain the extreme cost difference between high- and low-fidelity values often observed in well cited papers promoting new acquisition functions [7; 19]. See Figure 8 in the appendix for cost-correlation studies for the MF-TVR and MF-Custom formulations.

3.3 Acquisition Function Selection

The performance of MFBO is heavily influenced by the acquisition function used, as this decides which fidelity to evaluate next. Consequently, it is possible to improve the performance by choosing acquisition functions more appropriate for the problem context. Figure 3 compares the results across

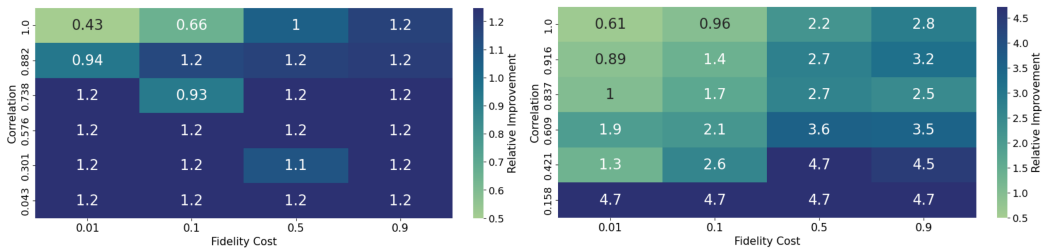


Figure 2: A heatmap for the MF-MES search algorithm showing how the correlation and cost of the low-fidelity data influences the optimization rate for (left) Problem 1 (RKHS) and (right) Problem 2 (6D negated Hartmann).

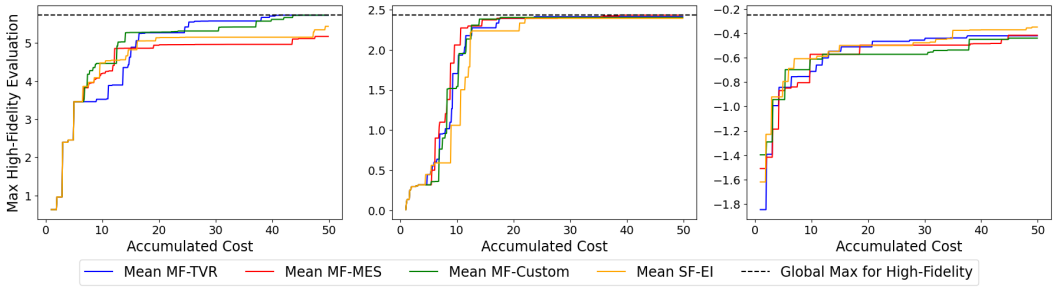


Figure 3: Comparison of each of the acquisition functions for Problems (left) 1 (RKHS), (middle) 2 (6D negated Hartmann), (right) 3 (organic photovoltaic molecules).

our four Problems when using MF-TVR, MF-MES, MF-Custom and SF-EI as the acquisition function. For Problems 1 and 2, MFBO consistently outperformed SFBO, however SF-EI performed better for the 5 runs for Problem 4. Interestingly, we observed in Figure 1 how MF-MES outperformed SF-EI with Problem 1 in obtaining the optimum for a single experiment, but see that when averaging over 5 runs, SF-EI outperformed MF-MES. Clearly, MFBO is not always guaranteed to provide a better result. However, the question as to what is even meant by 'better' arises, and we discuss what metrics are best to use to measure performance in the next section.

4 Conclusions and Outlook

We have explored the use of MFBO for chemical discovery. We found that MFBO outperformed SFBO in only two of our four problem datasets. These initial results were surprising, as we expected MFBO to, at worst, perform similarly to SFBO. We investigated this result firstly by exploring the impact of the low-fidelity data cost and its correlation to the high-fidelity data, finding that MFBO only consistently outperforms SFBO in cases where the cost of the low-fidelity data is below 10 to 100 times the high-fidelity one and with a high correlation above 0.9. This requirement limits the application of MFBO for chemical discovery, and should be addressed before its widespread implementation. Secondly, we explored the use of different acquisition functions. We considered three different acquisition functions and found that their performance was problem specific. This suggests the acquisition function should be carefully adopted for each case explored. We note that we did not explicitly consider the role of GP hyperparameters in this work; indeed, additional consideration is required here, as the GP hyperparameters are known to impact the optimization performance.

Simply measuring how efficiently the optimization process locates the maximum is not always a representative metric of the algorithm's performance, since in key applications, such as molecular discovery, the objective is not only to find the global maximum, but to seek multiple high-performing candidates. The metrics of *instantaneous regret* and *cumulative regret* are useful here, these can be used to measure how consistently high-performing selected candidates are, even once the optimum has been discovered. However, such metrics are not designed for the multi-fidelity setting, as regret simply takes into account an evaluation's distance from the high-fidelity optimum. The application

to a low-fidelity evaluation is thus less clear. Therefore, we suggest a custom metric of 'cumulative regret per high-fidelity evaluation' ($CRHF$), where:

$$CRHF = \sum_i \frac{\rho(i)}{\sum_{j=0}^{i} HF(j)} \tag{2}$$

so that the budget is divided into intervals i, $\rho(i)$ takes the most recent high-fidelity evaluation at i, and $\sum_{j=0}^{i} HF(j)$ is the number of high-fidelity evaluations up to i. When a low-fidelity value is selected, the metric takes the previous high-fidelity evaluation; however, to penalize a technique that simply takes low-fidelity evaluations repeatedly, the value is divided by how many high-fidelity values have been evaluated so far. See Figures 9 and 10 in the appendix for more results.

Code and Data Availability

The code for applying the search-algorithms to the RKHS and Hartmann functions, as well as to the COF dataset created by Gantzler et al., is available at `github.com/kernelCruncher/MFBO`. The code used for Problem 4 is available at `github.com/mohammedazzouzi15/STK_search/tree/master`.

References

R. Garnett, *Bayesian Optimization*, Cambridge University Press, Cambridge, 2023.

Y. Wu, A. Walsh and A. M. Ganose, *Digital Discovery*, 2024, **3**, 1086–1100.

H. Zhai and J. Yeo, *ACS Biomaterials Science & Engineering*, 2023, **9**, 269–279.

A. Gaonkar, H. Valladares, A. Tovar, L. Zhu and H. El-Mounayri, *Electronic Materials*, 2022, **3**, 201–217.

B. Jiang and X. Wang, *IEEE Control Systems Letters*, 2022, **6**, 1682–1687.

A. Deshwal, C. M. Simon and J. R. Doppa, *Molecular Systems Design & Engineering*, 2021, **6**, 1066–1086.

N. Gantzler, A. Deshwal, J. R. Doppa and C. M. Simon, *Digital Discovery*, 2023, **2**, 1937–1956.

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.

F. Strieth-Kalthoff, H. Hao, V. Rathore, J. Derasp, T. Gaudin, N. H. Angello, M. Seifrid, E. Trushina, M. Guy, J. Liu, X. Tang, M. Mamada, W. Wang, T. Tsagaantsooj, C. Lavigne, R. Pollice, T. C. Wu, K. Hotta, L. Bodo, S. Li, M. Haddadnia, A. Wołos, R. Roszak, C. T. Ser, C. Bozal-Ginesta, R. J. Hickman, J. Vestfrid, A. Aguilar-Granda, E. L. Klimareva, R. C. Sigerson, W. Hou, D. Gahler, S. Lach, A. Warzybok, O. Borodin, S. Rohrbach, B. Sanchez-Lengeling, C. Adachi, B. A. Grzybowski, L. Cronin, J. E. Hein, M. D. Burke and A. Aspuru-Guzik, *Science*, 2024, **384**, eadk9227.

M. Malu, G. Dasarathy and A. Spanias, 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, 2021, pp. 1–8.

D. Weininger, *Journal of Chemical Information and Computer Sciences*, 1988, **28**, 31–36.

M. Krenn, F. Häse, A. Nigam, P. Friederich and A. Aspuru-Guzik, *Machine Learning: Science and Technology*, 2020, **1**, 045024.

P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer and P. Friederich, *Commun. Mater.*, 2022, **3**, 1–18.

D. S. Wigh, J. M. Goodman and A. A. Lapkin, *WIREs Computational Molecular Science*, 2022, **12**, e1603.

L. David, A. Thakkar, R. Mercado and O. Engkvist, *J. Cheminf.*, 2020, **12**, 1–22.

E. Daxberger, A. Makarova, M. Turchetta and A. Krause, Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 2020, pp. 2633–2639.

H. Zhang, W. W. Chen, A. Iyer, D. W. Apley and W. Chen, *Uncertainty-Aware Mixed-Variable Machine Learning for Materials Design*, 2022, https://www.researchsquare.com/article/rs-1987975/v1.

A. Schoepfer, J. Weinreich, R. Laplaza, J. Waser and C. Corminboeuf, *Cost-Informed Bayesian Reaction Optimization*, 2024, https://chemrxiv.org/engage/chemrxiv/article-details/66220e8a21291e5d1d27408d.

C. Fare, P. Fenner, M. Benatan, A. Varsi and E. O. Pyzer-Knapp, *npj Computational Materials*, 2022, **8**, 257.

A. Tran, J. Tranchida, T. Wildey and A. P. Thompson, *The Journal of Chemical Physics*, 2020, **153**, 074705.

Z. Zanjani Foumani, M. Shishehbor, A. Yousefpour and R. Bostanabad, *Computer Methods in Applied Mechanics and Engineering*, 2023, **407**, 115937.

A. Palizhati, M. Aykol, S. Suram, J. S. Hummelshøj and J. H. Montoya, *Multi-fidelity Sequential Learning for Accelerated Materials Discovery*, 2021, https://chemrxiv.org/engage/chemrxiv/article-details/60c756c60f50dbb7f939813f.

A. Palizhati, S. B. Torrisi, M. Aykol, S. K. Suram, J. S. Hummelshøj and J. H. Montoya, *Scientific Reports*, 2022, **12**, 4694.

J. T. Springenberg, A. Klein, S. Falkner and F. Hutter, Advances in Neural Information Processing Systems, 2016.

Y. L. Li, T. G. J. Rudner and A. G. Wilson, *A Study of Bayesian Neural Network Surrogates for Bayesian Optimization*, 2024, http://arxiv.org/abs/2305.20028, arXiv:2305.20028 [cs, stat].

R.-R. Griffiths, L. Klarner, H. B. Moss, A. Ravuri, S. Truong, S. Stanton, G. Tom, B. Rankovic, Y. Du, A. Jamasb, A. Deshwal, J. Schwartz, A. Tripp, G. Kell, S. Frieder, A. Bourached, A. Chan, J. Moss, C. Guo, J. Durholt, S. Chaurasia, F. Strieth-Kalthoff, A. A. Lee, B. Cheng, A. Aspuru-Guzik, P. Schwaller and J. Tang, *GAUCHE: A Library for Gaussian Processes in Chemistry*, 2023, http://arxiv.org/abs/2212.04450, arXiv:2212.04450 [cond-mat, physics:physics].

M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson and E. Bakshy, *BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization*, 2020, http://arxiv.org/abs/1910.06403, arXiv:1910.06403 [cs, math, stat].

J. Wu, S. Toscano-Palmerin, P. I. Frazier and A. G. Wilson, *Practical Multi-fidelity Bayesian Optimization for Hyperparameter Tuning*, 2019, http://arxiv.org/abs/1903.04703, arXiv:1903.04703 [cs, math, stat].

S. Takeno, H. Fukuoka, Y. Tsukada, T. Koyama, M. Shiga, I. Takeuchi and M. Karasuyama, *Multi-fidelity Bayesian Optimization with Max-value Entropy Search and its parallelization*, 2020, http://arxiv.org/abs/1901.08275, arXiv:1901.08275 [cs, stat].

Gencoa Ltd and J. Brindley, author, 2022.

Y. Assael, *iassael/bo-benchmark-rkhs*, 2019, https://github.com/iassael/bo-benchmark-rkhs, original-date: 2014-10-18T14:58:36Z.

V. Picheny, T. Wagner and D. Ginsbourger, *Structural and Multidisciplinary Optimization*, 2013, **48**, 607–626.

L. Turcani, *lukasturcani/stk*, 2024, https://github.com/lukasturcani/stk, original-date: 2018-03-18T20:57:46Z.

C. Bannwarth, E. Caldeweyher, S. Ehlert, A. Hansen, P. Pracht, J. Seibert, S. Spicher and S. Grimme, *WIREs Computational Molecular Science*, 2021, **11**, e1493.

C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, MIT Press, Cambridge, Mass, 2006.

K. Wang, G. Pleiss, J. Gardner, S. Tyree, K. Q. Weinberger and A. G. Wilson, Advances in Neural Information Processing Systems, 2019.

P. Mikkola, J. Martinelli, L. Filstroff and S. Kaski.

S. Riniker and G. A. Landrum, *Journal of Chemical Information and Modeling*, 2015, **55**, 2562–2574.

S. Grimme, *J. Chem. Theory Comput.*, 2019, **15**, 2847–2862.

S. Grimme and C. Bannwarth, *J. Chem. Phys.*, 2016, **145**, 054103.

K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko and K.-R. Müller, *The Journal of Chemical Physics*, 2018, **148**, 241722.

A Appendix

A.1 Dataset Distributions

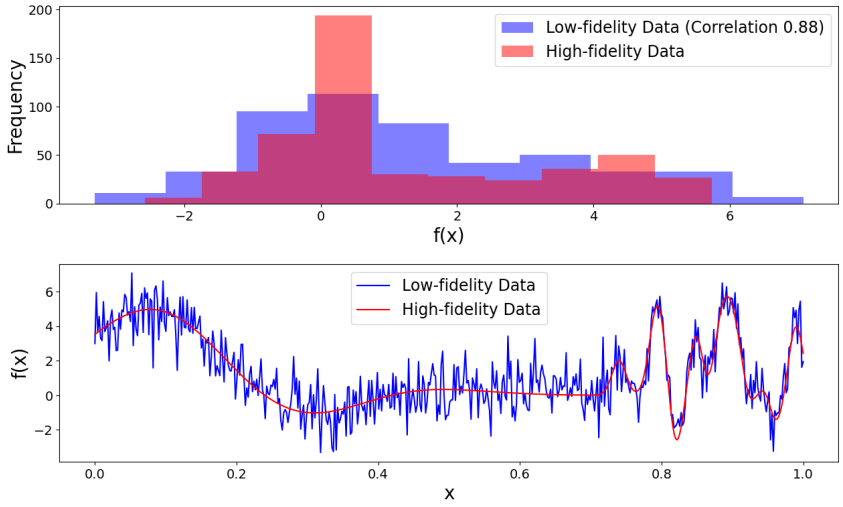


Figure 4: High- and low-fidelity data for the RKHS function and their distributions (Problem 1).

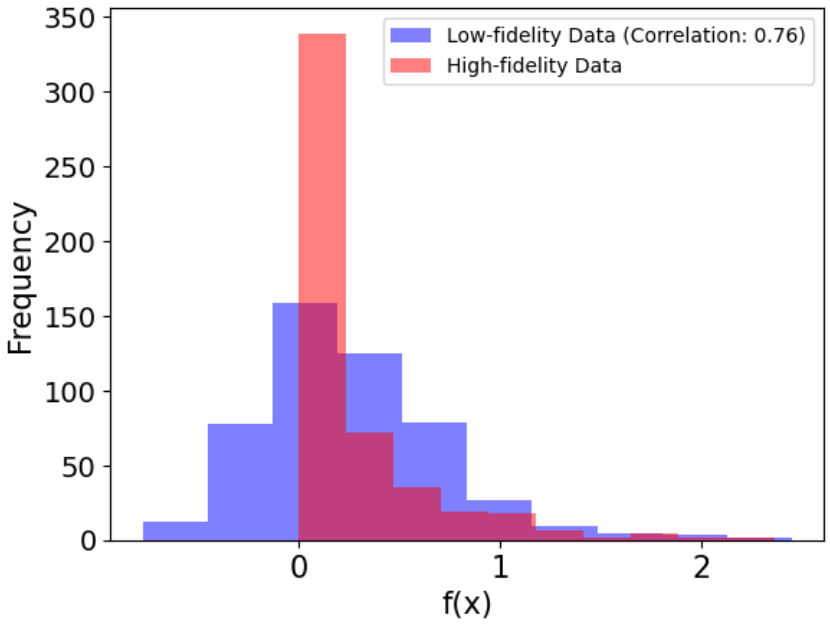


Figure 5: Histogram of 6D Hartmann evaluations (Problem 2).

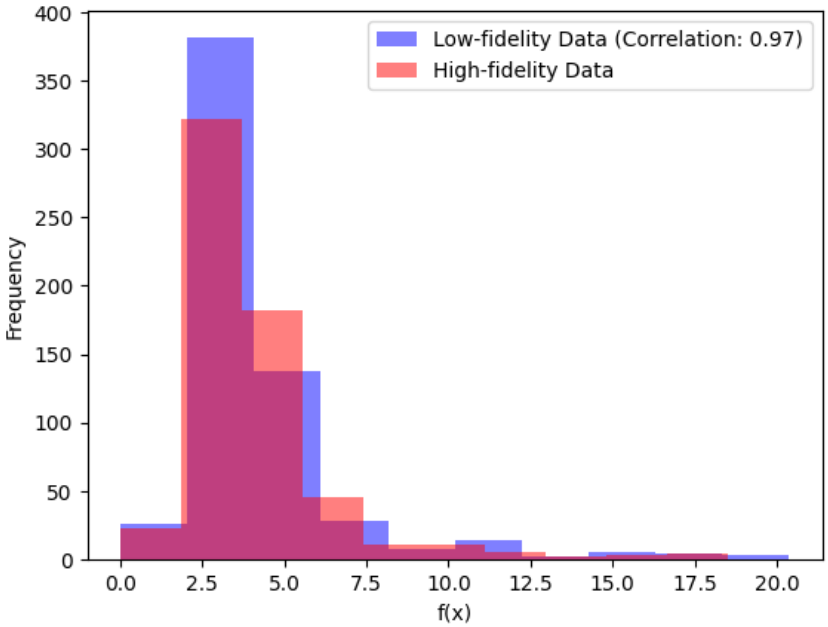


Figure 6: Histogram of COF selectivities (Problem 3).

A.2 BO and MFBO details

The BO algorithm has three components: the objective function, the surrogate model, and the acquisition function. The objective function is the function we are seeking to optimize and is considered a "black-box", i.e. its properties, such as gradient, are unknown and therefore less tractable to traditional techniques, like gradient-descent. The function is also considered expensive to evaluate (i.e. financially, computationally, or temporally) in some way and therefore must be invoked as few times as possible in the quest to find the the optimum.

The surrogate model is the approximate function the algorithm builds to model the objective function, and is cheaper to evaluate. Any model can be used as the surrogate, providing it encodes some level of uncertainty, which is critical to later stages of the algorithm. Traditionally, GPs have been used [26], but Bayesian neural networks are also possible [24; 25]. It should be stressed that GPs are best suited to small datasets, less than 10,000 training points, due to the cubic problem [35] (i.e. $O(n^3)$ computations for n training points); however, recent progress has been made to address this [36]. We adopt GPs and discuss the technical details below.

A GP is a stochastic process where every random variable is Gaussian, and moreover any finite collection of random variables is a multivariate Gaussian distribution, such that

$$[X_1, X_2, ..., X_n]^T \sim N(\vec{\mu}, \Sigma), \tag{3}$$

with $\vec{\mu}$ a vector representing the means of the finite collection of random variables, X_1, ..., X_n, and Σ defined as the covariance matrix. The covariance matrix is defined by the covariance function (also known as the kernel), K, and captures the dependency between the random variables as they are not necessarily independent, and is crucial for communicating to the model how close two random variables are, and therefore how similar their distributions should be. In particular, $\Sigma_{ij} = K(X_i, X_j)$. Examples of covariance functions include the RBF and Matern kernels (see reference [1] for a more detailed discussion). Furthermore, if we have D observations, $(x_1, ..., x_D)$, then we can compute the distribution of an unobserved value, x_*, via the equation:

$$f(x_*)|(f(x_1) = y_1, ..., f(x_D) = y_D) \sim N(k_*^T \Sigma^{-1} \vec{y}, K(x_*, x_*) - k_*^T \Sigma^{-1} k_*), \tag{4}$$

where $\vec{y} = [y_1, ..., y_D]^T$ is the vector of observations, $k_* = [K(x_1, x_*), ..., K(x_D, x_*)]^T$, with $K(\cdot, \cdot)$ the covariance function and Σ the covariance matrix described earlier. The idea is that each of the observations contributes to the distribution of the unobserved value x_*, via the covariance function.

The acquisition function is applied to the surrogate model to determine which point, or points, to ask the objective function to evaluate next, and assigns each point in the search-space a score. There are many different acquisition functions, such as Expected Improvement, Upper-Confidence Bound, Probability of Improvement, Knowledge Gradient and Maximum Entropy Search. Although defined differently, they are all trying to balance some notion of exploration (i.e. investigating those regions of the search-space with high uncertainty) against exploitation (i.e. remaining near those regions of the search-space with a high mean). Invariably, some acquisition functions are greedier than others, and the best to use depends not only on the search-space, but also the desired outcome of the experiment. Note that BO involves two distinct optimization problems: the outer optimization which is the original problem of optimizing the objective function, and the inner optimization, which is optimizing the acquisition function to determine the next point to evaluate. Usually, the acquisition function is defined in such a way as to make its optimization more computationally tractable; however, it should be stressed that this part of the algorithm can contribute to the computational cost.

MFBO is similar in design to the single-fidelity case discussed above: namely, both include surrogate models and acquisition functions. However, MFBO includes additional data-points of different fidelities that can be utilised in the pursuit of optimising for the target property at the desired fidelity. Consequently, the aim is no longer to reach the optimum in the fewest number of iterations, but rather to do so whilst exhausting the smallest budget. Lower-fidelity evaluations usually incur a lower cost, so providing there is some correlation between these and the high-fidelity target it is possible to optimize the objective function more economically. As a more concrete example, consider optimising for an electronic property in materials-design, where the high-fidelity data is experimental evaluation, and the lower-fidelity data is a computational chemistry calculation, either ab-initio or empirical. The former is extremely time-consuming and expensive, so any relationships in the domain that can be learned from computation would mitigate the experimental burden.

One further distinguishing feature from the single-fidelity case is that the input to the acquisition function includes the fidelity. This is because the output must also specify the fidelity, since it is quite possible that the same point in the search-space brings a different value to the optimization process depending on the fidelity it is evaluated at. This difference in behaviour requires new acquisition functions to be defined to dictate how the additional information should be utilised.

For a detailed description of the multi-fidelity kernel used in this work we refer the reader to section F in 37.

A.3 RKHS function

The RKHS function is a standard benchmarking function for optimization algorithms. This presents a complex, high-dimensional parameter space to search, while still being analytically tractable for conventional methods. This is powerful for comparison studies like the one presented in this work.

See Assael (2019) for the definition of the function, which can be described using two Squared Exponential kernels. The known solution is located at the point

$$x = 0.892, f(x) = 5.734.$$

A.4 `STK_search` dataset

The molecules in `STK_search` are composed of 6 building-blocks, or units, each of which has 306 different candidate structures. This creates a combinatorial space of $306^6 = 8 \times 10^{14}$ possible oligomers to be explored. From this space we only consider 45,000 molecules with 30,000 randomly selected and 15,000 generated through trying different optimisation algorithms. Each unit is fed into a deep learning model that embeds its features, such as HOMO, LUMO, excited state energy, etc, into 12-dimensional vector space, thus causing the whole molecule to be represented by a 72-dimensional vector. For the `STK_search`, the BO proceeds similarly to what is described in the main text, namely the GP is seeded with an initial sample, which an acquisition function then uses to determine the most valuable elements to evaluate next. However, due to the size of the space it no longer becomes computationally feasible to apply the acquisition function to every element, and instead, an evolutionary algorithmic approach is applied that combines the building-blocks of the strongest candidates, so far, whilst also injecting randomness, via mutations, to ensure a wide, albeit incomplete, search of the space. As the name suggests, evolutionary algorithms are inspired by biology, specifically natural selection, where the unit-like structure resembles a sequence of genes competing for dominance.

We optimize the molecules to align with the specific properties of a molecular acceptor, namely ionisation potential and complimentary absorption. The property of the molecule to be predicted, therefore, is the combined-target function, F_{comb}, defined as

$$F_{comb} = -|E_{S1} - 3| - |IP - 5.5| + \log(f_{osc,S1}), \tag{5}$$

where IP is the ionization potential, E_{S1} is the first excited state energy, and $f_{osc,S1}$ is the oscillator strength of the first excited state. To calculate these properties, we used stk to generate initial geometries for the molecules, followed by the use of the Experimental-Torsion basic Knowledge Distance Geometry (ETKDG) approach in stk/RDkit to generate a first geometry.[38] Then, we optimised the geometry of the lowest energy conformer found using GFN2-XTB [39] and calculated the ionisation potential and electron affinity using the IPEA option in XTB. The optical properties of the molecules were calculated using sTDA-XTB. [40] The lower-fidelity data is produced by the predictions of a machine-learning model trained on the same dataset. Here we used Schnet as the surrogate model[41]. The low- and high-fidelity data are available at `github.com/kernelCruncher/MFBO`. We assigned an initial cost of 0.1, although there is some discussion surrounding the significance of this cost-value and its impact to the MFBO process. Indeed, we posit that the selection of fidelity values should be an ongoing area of research, with design-principles yet unknown and problem-specific.

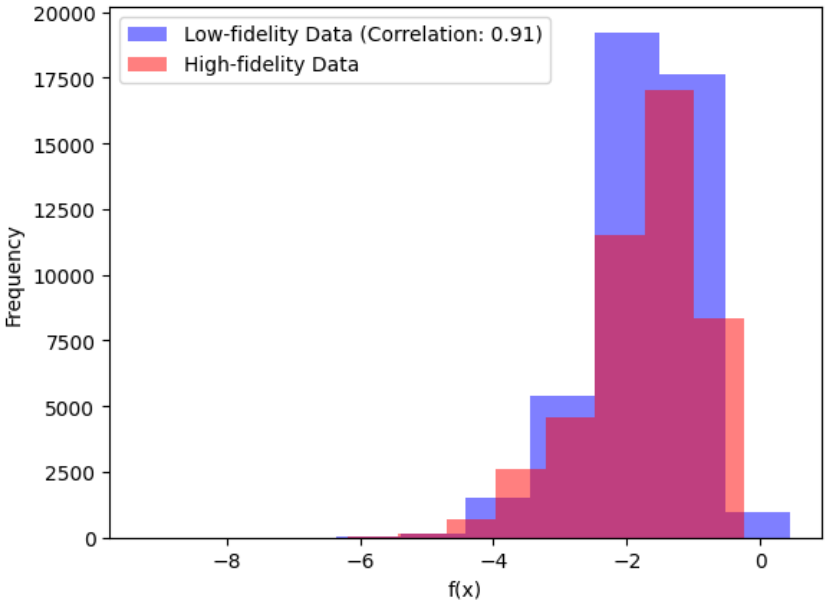
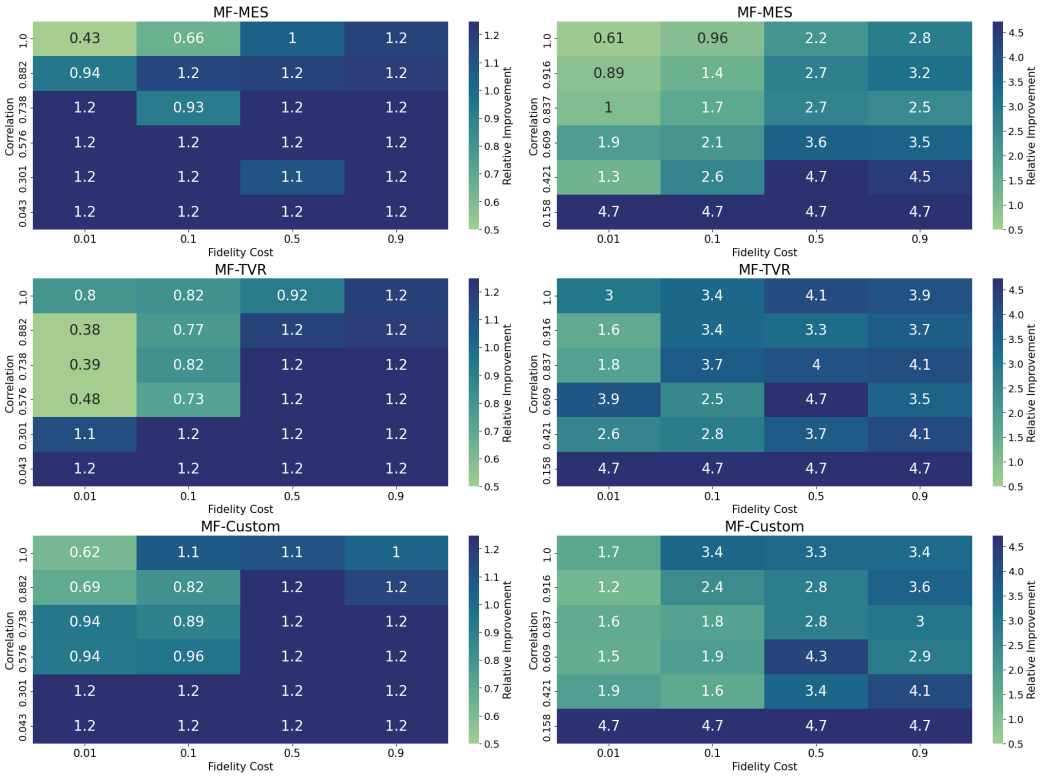


Figure 7: Histogram of combined-target evaluations for the organic photovoltaic molecules (Problem 4).

A.5 Fidelity Cost and Correlation Heatmaps



(a) RKHS (b) 6D negated Hartmann

Figure 8: A heatmap for the MF-MES, MF-TVR and MF-Custom acquisition functions illustrating how the correlation and cost of of the low-fidelity data influences the rate of optimization for (a) RKHS (Problem 1) and (b) 6D negated Hartmann (Problem 2).

A.6 Custom Metrics

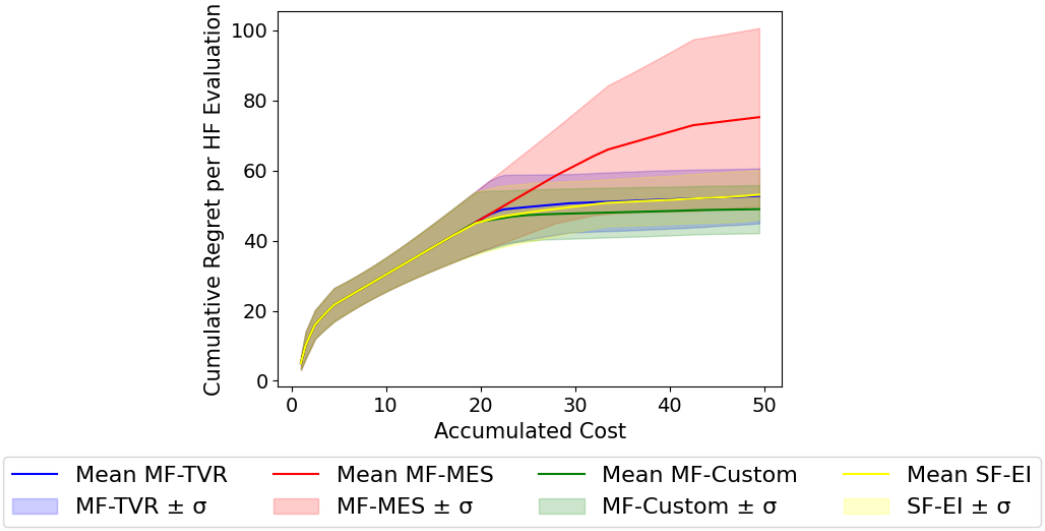


Figure 9: A comparison of the acquisition functions in a single plot for RKHS (Problem 1), using cumulative regret per high-fidelity evaluation. Each experiment was seeded with 5 initial samples (both high- and low-fidelity evaluations), afforded a budget of 50, and had a domain-size of 500. The results are averaged over 5 runs for each search-algorithm.

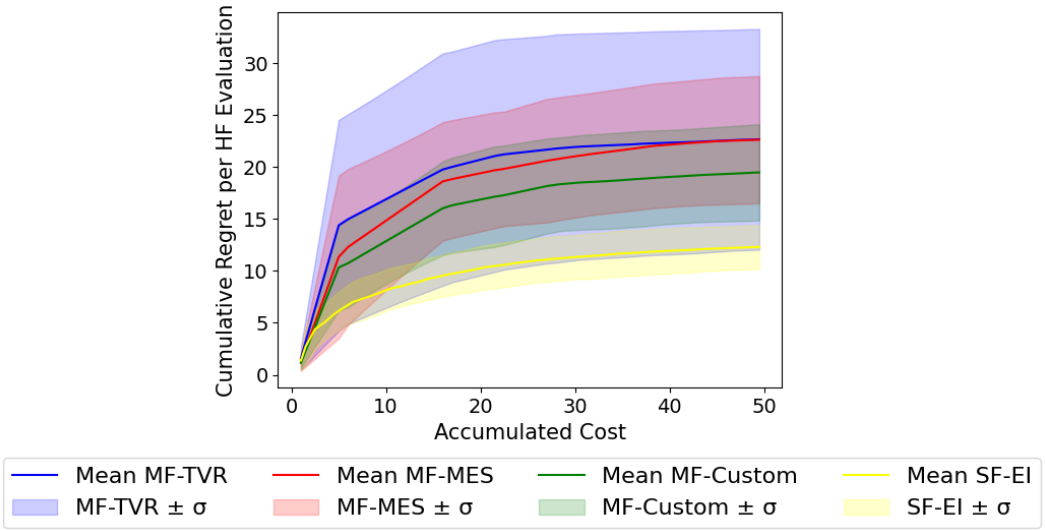


Figure 10: A comparison of the acquisition functions in a single plot for Problem 4, using cumulative regret per high-fidelity evaluation. Each experiment was seeded with 25 initial samples (both high- and low-fidelity evaluations), afforded a budget of 50, and had a domain-size of 44928. The results are averaged over 5 runs for each search-algorithm.