Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Comparison of Multi-Fidelity Modelling Methods for Bayesian Optimization

Stefan Tönnis[a], Luise F. Kaven[a], Eike Cramer[a,*]

[a] Process Systems Engineering (AVT.SVT), RWTH Aachen University, 52074 Aachen, Germany
* Corresponding Author: eike.cramer@avt.rwth-aachen.de

ABSTRACT

In process systems engineering (PSE), obtaining accurate process models for optimization can be expensive and time-consuming. Black-box Bayesian Optimization (BO) with Gaussian process (GP) surrogates offers a promising approach. However, full black-box optimization neglects valuable prior knowledge, which could otherwise improve the optimization process. This work explores methods of integrating prior knowledge in the form of low-fidelity data into BO by evaluating these methods on synthetic multi-fidelity test functions. Our results highlight possibilities for improved convergence of the BO optimization. However, our work further highlights potential pitfalls of these multi-fidelity models, such as bias, convergence to local optima, and overfitting on low-fidelity data. Hence, leveraging low-fidelity data in multi-fidelity models can improve BO convergence, but there are instances where the algorithms are more susceptible to failure.

Keywords: Process Design, Optimization, Machine Learning, Numerical Methods

INTRODUCTION

Process models are not always available and can be prohibitively expensive to obtain for model-based optimization. Hence, the process systems engineering (PSE) community has gained an interest in Bayesian Optimization (BO). BO uses Gaussian process (GP) surrogate models as a probabilistic approximation of objective functions. The BO algorithm iteratively queries the black-box objective with experimental conditions obtained from optimizing over the so-called acquisition function [1]. Although BO is generally known as sample-efficient, treating chemical engineering design problems as fully black-box problems can still be prohibitively expensive, particularly for high-cost technical-scale experiments with many degrees of freedom (DoF). At the same time, the past 150 years of chemical engineering research provide extensive knowledge, modeling, and databases that are neglected by black-box algorithms such as BO. Hence, we investigate different possibilities of including such prior knowledge in the BO algorithm.

Other works from the PSE community investigate options to combine BO with known model equations, e.g., in compositions of GPs with mechanistic equations [2, 3]. In this work, however, we consider the case of available data, e.g., from low-cost simulations, approximate models, or similar processes.

One widely known option to include such prior knowledge in BO is prior mean modeling [4], where the user complements the BO algorithm with an initial guess, i.e., a prior mean. Such prior means may be derived from a scientist's intuition or fitted to available lower-fidelity data. A similar yet structurally different alternative are multi-fidelity GPs designed to handle lower-fidelity data [5]. These multi-fidelity models enhance the model's predictive capabilities by transferring knowledge from low-fidelity observations to the high-fidelity regime. Prominent methods are multi-task and multi-fidelity GPs utilizing special kernels to model the relations between different fidelities or the recursive GP model introduced by Kennedy and O'Hagen [6, 7].

In this work, we compare the usage of different multi-fidelity and prior mean modeling techniques for BO design problems. We focus on improving BO convergence in the high-fidelity regime, utilizing low-fidelity data as a supplementary source of information. Our analysis shows that utilizing lower-fidelity data in BO can generally improve the convergence of the BO optimization. However, our analysis also highlights some potential pitfalls and causes for failure, such as convergence to

local optima, overfitting, and model training errors. For this conference proceeding, we restrict our analysis to established multi-fidelity test functions, e.g., Forrester, Rosenbrock, and Rastrigin test functions.

BAYESIAN OPTIMIZATION

Black-box optimization refers to optimization problems where the objective function $f(\mathbf{x})$ lacks an analytical expression or is expensive to evaluate. The general form of a black-box optimization problem is:

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{1}$$

Here, $\mathbf{x}$ is the vector of DoF from a feasible domain $\mathcal{X} \subseteq \mathbb{R}^{\mathrm{d}}$, and $f(\mathbf{x})$ is the objective function to be minimized. BO is a widely adopted approach for solving such problems, leveraging probabilistic surrogate models to approximate the objective function, enabling efficient exploration and exploitation of the search space while keeping the number of evaluations low.

GPs are a popular choice for the probabilistic surrogate model for BO due to their data efficiency and inherent uncertainty quantification. GPs are an extension of a multivariate normal distribution to infinite dimensions, i.e., continuous functions, with the prior distribution $p(f)$ of the objective function f defined as:

$$p(f) = \mathcal{N}\left(f; \mu(\mathbf{x}), k(\mathbf{x},\mathbf{x}') = \mathcal{GP}\big(\mu(\mathbf{x})\big), k(\mathbf{x},\mathbf{x}')\right) \tag{2}$$

Here, $\mu(\mathbf{x})$ is the mean function representing the expected value of the objective function f, and $k(\mathbf{x},\mathbf{x}')$ is the covariance function describing the relation between two arbitrary vectors $\mathbf{x}$ and $\mathbf{x}'$. The joint distribution of the prior belief f and exact observations $\mathbf{y}$ is Gaussian:

$$p(f,\mathbf{y}) = \mathcal{GP}\left(\begin{bmatrix}\mu(\mathbf{x})\\ \mathbf{m}\end{bmatrix}, \begin{bmatrix}k(\mathbf{x},\mathbf{x}') & k(\mathbf{x},\mathbf{X})^{\mathrm{T}}\\ k(\mathbf{X},\mathbf{x}) & \mathbf{K}(\mathbf{X},\mathbf{X})\end{bmatrix}\right), \tag{3}$$

The vector of means of the observations, $\mathbf{m}$, is typically set directly to the observations. For the case of exact observations, this simplifies to $\mathbf{y} = \{f(\mathbf{x}_i)\}_{i=1}^{N}$, where N denotes the number of observations. The matrix $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ represents the corresponding set of data points in the input dimension and $\mathbf{K}(\mathbf{X},\mathbf{X})$ corresponds to the covariance matrix evaluated at these data points, using a kernel function $k(\cdot,\cdot)$. Conditioning the GP prior $p(f)$ on the noise-free observation data $\mathcal{D} = \{\mathbf{x}_i, f(\mathbf{x}_i)\}_{i=1}^{N}$ yields the posterior $p(f|\mathcal{D})$, which is also Gaussian:

$$p(f|\mathcal{D}) = \mathcal{GP}(f; \mu_{\mathcal{D}}(\mathbf{x}), k_{\mathcal{D}}(\mathbf{x},\mathbf{x}')) \tag{4}$$

The posterior mean and covariance functions read:

$$\mu_{\mathcal{D}}(\mathbf{x}) = \mu(\mathbf{x}) + K(\mathbf{x},\mathbf{X})K(\mathbf{X},\mathbf{X})^{-1}\big(\mathbf{m} - \mu(\mathbf{X})\big) \tag{5}$$

$$k_{\mathcal{D}}(\mathbf{x},\mathbf{x}') = k(\mathbf{x},\mathbf{x}') - k(\mathbf{x},\mathbf{X})K(\mathbf{X},\mathbf{X})^{-1}k(\mathbf{X},\mathbf{x}) \tag{6}$$

Since the surrogate model is probabilistic, it allows the user to formulate an expression of a preference over the uncertain outcomes, also known as an acquisition function $\alpha : \mathcal{X} \to \mathbb{R}$. Thus, the inference of the next observation location $\hat{\mathbf{x}}$ boils down to maximizing the acquisition function:

$$\hat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmax}}\, \alpha(\mathbf{x},\mathcal{D}) \tag{7}$$

One of the most common acquisition functions is the expected improvement acquisition function, which is defined as:

$$\mathrm{EI}_{y^*}(\mathbf{x}) = \mathbb{E}_{f(\mathbf{x})}[\max(0, f(\mathbf{x}) - y^*)]\,, \tag{8}$$

where the incumbent $y^* = \max \mathbf{y}$ is the best function value observed so far [8].

Bayesian optimization with available lower-fidelity data

In many chemical engineering problems, data often comes from simulations or experiments that can be performed at varying levels of detail, precision, or cost. These levels of detail are commonly referred to as fidelities. Each level of fidelity can be represented by a different function $f_t(\mathbf{x})$, where $t = 1, \ldots, T$ indicates the fidelity level, with $t = T$ representing the highest fidelity. Each fidelity level provides a set of data points, $\mathcal{D}_t = \{(\mathbf{x}_i, f_t(\mathbf{x}_i))\}_{i=1}^{N_t}$ where $\mathbf{x}_i$ represents DoF and $f_t(\mathbf{x}_i)$ the corresponding function evaluations, with N_t denoting the number of observations in the t-th fidelity.

The goal of multi-fidelity optimization is to efficiently determine the optimal choice of the DoF $\mathbf{x}^*$ that minimizes the highest-fidelity function $f_T(x)$. This is achieved by leveraging data from the lower-fidelity levels $\{\mathcal{D}_t\}_{t=1}^{T-1}$, which are typically less expensive to compute, to inform and reduce the need for evaluations of $f_T(\mathbf{x})$. The challenge lies in effectively combining information from these fidelities to balance accuracy and computational efficiency.

While multi-fidelity optimization often incorporates a cost-aware perspective, assigning evaluation costs to fidelities and using a joint acquisition function to minimize expected cost, this work focuses solely on the highest-fidelity function. Here, lower-fidelity data is used to enhance the predictive model of $f_T(\mathbf{x})$, improving accuracy and reducing the number of high-cost evaluations required.

Prior mean modeling

The prior mean function $\mu(\mathbf{x})$ of a GP is typically assumed to be either zero or a constant value in order to avoid any unwanted bias on our decisions caused by spurious structure in the prior process [1, 9]. By specifying the prior mean function $\mu(\mathbf{x})$ of the prior belief of the objective function $p(f)$, high-level information about the objective function can incorporated into the BO process. The GP posterior of the objective function $p(f|\mathcal{D})$ is then updated by the available data but still inherits the

previously encoded knowledge.

A practical approach to integrating low-fidelity data into the prior mean of a Gaussian process involves training a regression model on the low-fidelity dataset. The resulting model is then utilized as the prior mean function for the Gaussian process. In this study, we consider a fifth-order polynomial regression with L2 regularization and a neural network. For the polynomial regression model, the prior mean of the GP is defined as:

$$\mu(\mathbf{x}) = \Phi(\mathbf{x})\mathbf{w} \tag{9}$$

Here $\Phi(\mathbf{x})$ represents the feature matrix, and $\mathbf{w}$ the respective weights of the polynomial regression model. The prior mean of the neural network is defined as:

$$\mu(\mathbf{x}) = \mathrm{NN}(\mathbf{x}) \tag{10}$$

Here, $\mathrm{NN}(\boldsymbol{x})$ represents the neural network.

Recursive co-kriging model

The recursive co-kriging model, commonly referred to as the auto-regressive (AR1) model, was first introduced by Kennedy and O'Hagen [6] and further elaborated upon by Gartiet [7]. This model is widely used for multi-fidelity modeling and leverages an auto-regressive structure to represent dependencies across fidelity levels. It assumes a Markov property, wherein each fidelity level is dependent only on the immediately preceding level. The general formulation of the AR1 model for T fidelity levels can be expressed as:

$$f_t(\mathbf{x}) = \rho_{t-1} f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}), \qquad t = 2, \dots, T \tag{11}$$

Here, $f_t(\mathbf{x})$ represents the GP associated with the t-th fidelity level, while $f_{t-1}(\mathbf{x})$ is the GP conditioned on the data from the preceding fidelity level. The parameter ρ_{t-1} serves as a scaling factor that quantifies the influence of the $(m-1)$-th fidelity level on the current fidelity level. The term $\delta_t(\mathbf{x})$ is a GP that models the residual errors between the high-fidelity data at the t-th level and the predictions provided by $f_{t-1}(\mathbf{x})$.

Due to the AR1 model's Markov property, the data points at a given fidelity level only influence the predictions of that level and all subsequent fidelity levels, but not the preceding fidelity levels. Consequently, the model is well-suited for datasets with such a hierarchical structure.

Multi-Fidelity Modeling

Unlike the recursive co-kriging model, which leverages an auto-regressive structure to model the fidelity relation sequentially, other multi-fidelity approaches adopt a joint modeling framework. A key challenge in such models lies in accurately defining and learning the cross-covariance structure between tasks or fidelities, which is critical for effective knowledge transfer.

Two notable examples of such an approach are the multi-task GP model formulation proposed by [10] and the multi-fidelity GP model formulation proposed by [11]. The multi-task model captures these relationships using a learnable inter-task covariance matrix. In contrast, the multi-fidelity model by [11] represents fidelity as a continuous parameter and employs a covariance function to describe inter-task dependencies. Both approaches rely on decomposing the covariance between outputs into components that reflect task/fidelity dependencies and input space similarities. This can be formally expressed as:

$$k_{multi}\big((\mathbf{x}, t), (\mathbf{x}', t')\big) = k_t(t, t') \cdot k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}') \tag{12}$$

or equivalently in tensor form:

$$k_{multi}\big((\mathbf{x}, t), (\mathbf{x}', t')\big) = \mathbf{K}_t \otimes k_{\mathbf{x}}(\mathbf{x}, \mathbf{x}') \tag{13}$$

Here, t and t' denote task or fidelity indices, k_t represents the covariance of tasks or fidelities, and k_x encodes input space similarity. The choice of representation depends on whether tasks are treated as discrete entities, e.g., in multi-task GPs, or part of a continuous fidelity spectrum, e.g., in multi-fidelity GPs.

RESULTS AND DISCUSSION

The different multi-fidelity BO methods are evaluated on optimization test problems, specifically on multi-fidelity versions of the Forrester, the Rosenbrock, shifted and rotated Rastrigin, and the Heterogeneous test functions. Both the shifted and rotated Rastrigin and the Heterogeneous function are multi-modal functions, i.e., these functions exhibit multiple local optima. The key characteristics are summarized in Table 1. For a detailed explanation of the test function, the reader is referred to the original source [12]. The different methods are compared to a baseline, i.e., a standard GP conditioned only on high-fidelity data. All evaluated methods are summarized in Table 2.

Table 1: Evaluated multi-fidelity test functions [12].

#	Function	# Dimensions	# Fidelities
1	Forrester	1	4
2	Rosenbrock	2	3
3	Rastrigin	2	3
4	Heterogeneous	1	2

The performance of the different methods is evaluated on the success rate of optimization runs. We evaluate the success of an optimization run using a convergence score. The convergence score is defined as:

$$\Phi_i = \frac{f_{max} - y_i^*}{f_{max} - f_{min}} \tag{14}$$

Here, f_{min} and f_{max} are the minimum (optimum) and maximum function values in the search domain, respectively.

Here, y_i^* is the incumbent from the methods in the i-th optimization step. The convergence score is then compared to a required convergence score $\phi \leq \phi_{req}$. If the required convergence score is reached, the optimization run will be deemed successful. Note that we can define the convergence score as we consider test functions with known optima.

Table 2: Evaluated multi-fidelity modeling methods.

#	Methods	Description
1	Baseline	GP conditioned on high-fidelity data
2	Prior Mean Polynomial	GP conditioned on high-fidelity data and using a polynomial regression model trained on low-fidelity data.
3	Prior Mean NN	GP conditioned on high-fidelity data and using a neural network regression model trained on low-fidelity data.
4	Multi-Fidelity GP	Multi-fidelity GP model as described in [11], conditioned on multi-fidelity data.
5	Multi-Task GP	Multi-task GP model as described in [10], conditioned on multi-fidelity data.
6	Recursive Co-Kriging	Recursive Co-Kringing model as described in [6], conditioned on multi-fidelity data.

All methods are run for a total number of 100 optimization runs, with varying initial data and with a maximum budget of 100 optimization steps. We use 10 initial data points per dimension for each low-fidelity function. Additionally, one data point per dimension in the target fidelity was considered as initial data. A small observation noise covariance prevents numerical instability, assuming exact observations otherwise. The logarithmic expected improvement is used as an acquisition function. The logarithm of the expected improvement is maximized to account for vanishing gradients in low-value regimes [8].

Figure 1 shows the evaluation results for the convergence score. Each subplot (1-4) corresponds to one optimization problem from Table 1, with the x-axis in logarithmic scale representing the number of iterations and the y-axis depicting the success rate, ranging from 0 to 100%. The success rate measures the proportion of the total optimization runs that successfully reached the desired performance threshold in each iteration step. The markers indicate failed optimization runs.

The Baseline achieves 100% success after 15, 31, 33, and 79 iterations for problems 1–4, respectively. For the Heterogeneous problem, 90% of runs converge by iteration 32, with a few requiring up to 79 iterations.

The Prior Mean Polynomial method slightly outperforms the Baseline for the Forrester problem, converging 1.5 iterations faster on average. It performs significantly better in the Rosenbrock problem during the first 15 iterations, achieving a 3.2 iteration faster average convergence while matching the Baseline for full convergence. Performance on the Rastrigin problem mirrors the Baseline, but for the Heterogeneous optimization problem, it falls short, with 32 runs failing to converge, achieving only a final success rate of 68%.

The Prior Mean Neural Network underperforms across all problems, with slower average convergence of 0.3, 2.9, 11.2, and 5.1 iterations, respectively, only considering successful runs. It fails to converge on three out of four optimization problems, with two failed runs for the Forrester problem, nine for the Rastrigin problem, and 34 for the Heterogeneous problem.

The Multi-Fidelity GP method encounters issues in the Forrester problem, with 23 runs failing due to model fitting errors (77% success rate). For successful runs, convergence is slower by 2.4 iterations. Meanwhile, the Multi-fidelity GP generally achieves faster convergence for problems 2–4, with average improvements of 4.6, 2.8, and 7.8 iterations, respectively. Nevertheless, one optimization trial in the Heterogeneous problem does not converge within the given iteration limit.

The Multi-Task GP excels in the Forrester problem, achieving 99% convergence within six iterations and full convergence in 11, outperforming the Baseline by 6.5 iterations on average. In the Rosenbrock problem, despite five model fitting errors lowering the success rate to 95%, successful runs converge 13.8 iterations faster. For the Rastrigin problem, 60% of runs converge within 21 iterations, averaging 3.5 iterations faster, but failed runs result in an overall slowdown of 2.1 iterations. In the Heterogeneous problem, the method achieves 75% success with an average speedup of 3.0 iterations, though eight runs fail to converge.

The Recursive Co-Kriging model demonstrates significantly faster convergence compared to the Baseline for the Forrester, Rosenbrock, and Heterogeneous problems, with average improvements of 5.8, 10.1, and 8.6 iterations. In the Rastrigin problem, 70% of runs converge 4.7 iterations faster, but the remainder either match or lag behind the Baseline, resulting in a marginal 0.3 iteration faster average overall.

The Prior Mean Polynomial regression model demonstrates that prior mean information can enhance convergence to the optimal solution, particularly for the Forrester and Rosenbrock optimization problems, where it outperforms the baseline. However, as seen in the Heterogeneous problem, the prior mean can also mislead the optimization. When convergence failed, excessive sampling at domain boundaries occurred, likely due to

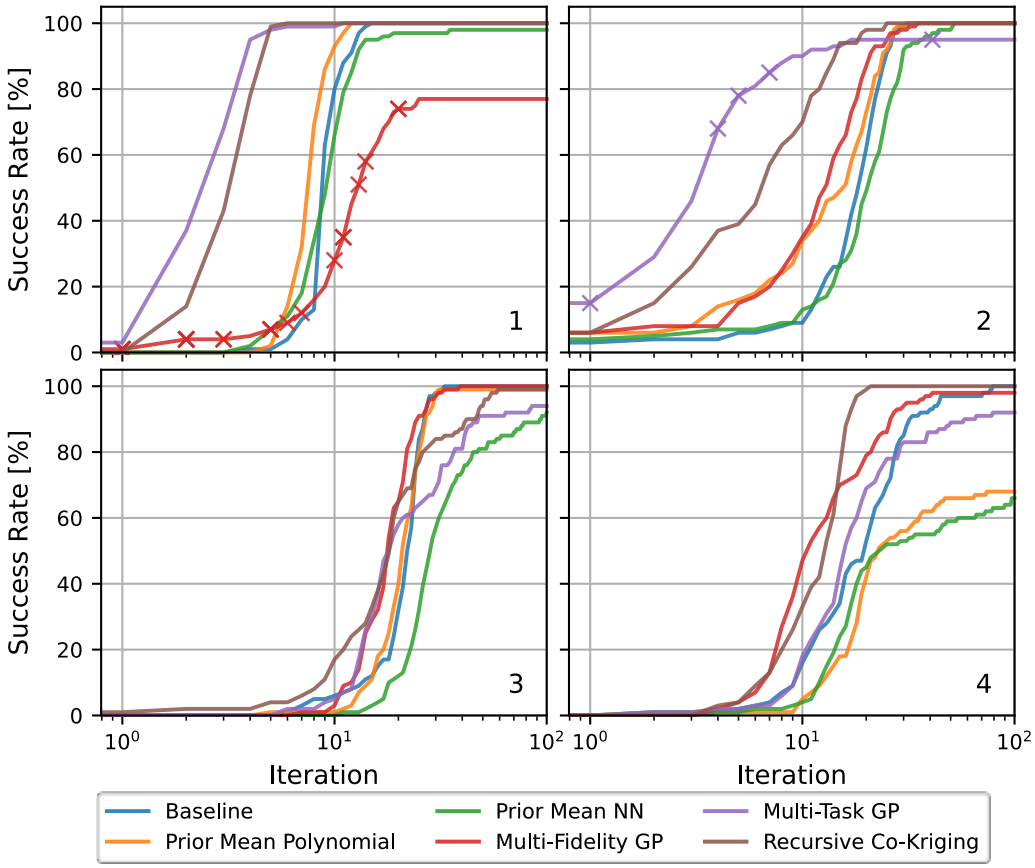

Figure 1: Success rate of the different multi-fidelity methods evaluated on the Forrester, Rosenbrock, Rastrigin and Heterogeneous multi-fidelity test problems (1-4). The success rate is plotted against the number of iterations for each method. The methods compared are: Baseline, Prior Mean Polynomial, Prior Mean NN, Multi-Task GP, Multi-Fidelity GP, and Recursive Co-Kriging. Markers indicate failed optimization runs.

unfavorable prior mean predictions leading to large lengthscale parameters, reduced predictive variance, and repeated boundary selections. In these instances, a near-linear structure of the polynomial regression prior biases the first BO step toward the domain boundary. A visualization of one of these instances can be seen in Figure 2. High regularization penalties further reinforce this tendency.

The cause of failure for the Neural Network regression model on the Heterogeneous test problem is similar to that of the Polynomial regression model. For the Rosenbrock test function, overfitting to the different fidelity observations contributes to the method's poor performance. For the Rastrigin test problem, the lower-fidelity functions penalize the region around the optimum, leading to repeated observations in areas of local optima.

Similar behavior is observed in the Multi-Task GP and Recursive Co-Kriging methods, for which the Multi-Task GP seems to generally struggle with the multi-modal test problems 3-4. The Multi-Fidelity GP method, on the other hand, outperforms the Baseline on test problems 2-4. Both the Multi-Task GP and the Multi-Fidelity GP tend to fail to converge during model training. This is likely due to their higher number of hyperparameters which complicates model training. Additionally, the Multi-Fidelity GP method assumes an ordering of the fidelity functions as the model assumes that the fidelity of the function is monotonic in the parameter [11]. If such a structure is not given, the model can fail during training due to an ill-conditioned covariance matrix. In summary, all multi-fidelity methods are prone to biases from poorly distributed initial low-fidelity samples, with the Multi-Fidelity GP being the most resilient. Among the considered methods, the Recursive Co-Kriging model is the most promising, consistently outperforming the Baseline on average across all optimization problems.

CONCLUSION

This paper evaluates several common multi-fidelity modeling methods using multi-fidelity test functions and

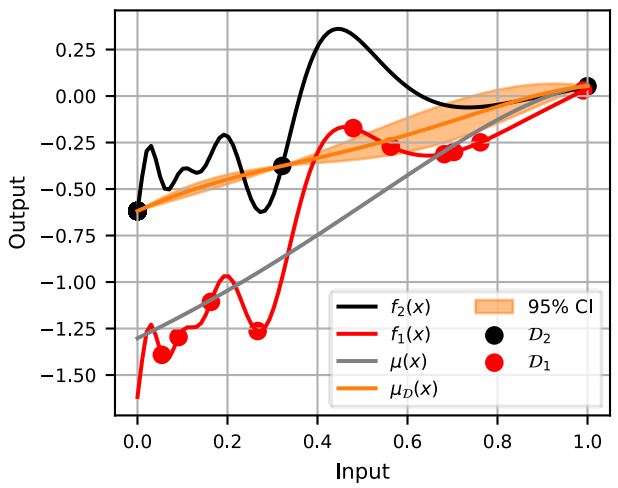

Figure 2. Failed optimization run of the Prior Mean method on the Heterogeneous test function showing the posterior GP prediction, conditioned on data points $\mathcal{D}_2$, using low-fidelity data $\mathcal{D}_1$ to train the polynomial regression mean $\mu(\mathrm{x})$, with the 95% confidence interval of the posterior GP indicated as CI.

identifies key pitfalls associated with these approaches. A fundamental limitation of all evaluated methods is their purely data-driven nature, making their performance highly dependent on data quality. If the available low-fidelity data is unfavorably distributed or biased, the optimization process may get misguided. This is especially evident for the prior mean modeling method, which, while being an easy-to-implement and potentially effective strategy, showed poor performance when using a Neural Network prior mean. The Neural Network severely overfitted the available data, encoding misleading information that negatively impacted optimization performance. Therefore, careful consideration is necessary when selecting and designing a prior mean model.

The more advanced multi-fidelity modeling methods may offer enhanced capability but are not exempt from the issue of misleading prior information, especially in the case of multi-modal objective functions. Among the evaluated methods, the Recursive Co-Kriging model demonstrated the most consistent performance, outperforming the baseline across all test problems, while the Multi-Fidelity GP proved the most resilient to the issue of misleading biases. To fully leverage the potential of multi-fidelity modeling, robust model fitting algorithms should be employed to improve reliability.

Finally, this study focuses on low-dimensional test functions. While the presented methods can be applied to high-dimensional problems, appropriate data scaling becomes crucial as dimensionality increases. Bayesian optimization often struggles in higher dimensions, but techniques like TURBO have been shown to improve performance [13].

REFERENCES

1. Garnett R. *Bayesian optimization*. Cambridge University Press, Cambridge, United Kingdom (2023).
2. Astudillo R, Frazier PI. Bayesian Optimization of Composite Functions doi:10.48550/arXiv.1906.01537 (2019) (Epub ahead of print) .
3. González LD, Zavala VM. BOIS: Bayesian Optimization of Interconnected Systems. *IFAC-PapersOnLine* 58(14), 446–451 (2024).
4. Chitre A, Cheng J, Ahamed S *et al.* pHbot: Self-Driven Robot for pH Adjustment of Viscous Formulations via Physics-informed-ML**. *Chemistry Methods* 4(2) (2024).
5. Wu J, Frazier PI. Continuous-Fidelity Bayesian Optimization with Knowledge Gradient (2017).
6. Kennedy M. Predicting the output from a complex computer code when fast approximations are available. *Biometrika* 87(1), 1–13 (2000).
7. Le Gratiet L, Garnier J (2012). *Recursive Co-Kriging Model for Design of Computer Experiments with multiple Levles of Fidelity*.
8. Ament S, Daulton S, Eriksson D, Balandat M, Bakshy E (2023). *Unexpected Improvements to Expected Improvement for Bayesian Optimization*.
9. Ath G de, Fieldsend JE, Everson RM. What do you mean? *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*. Presented at: *GECCO '20: Genetic and Evolutionary Computation Conference*. Cancún Mexico, 08 07 2020 12 07 2020.
10. Bonilla EV, Chai, Kian Ming A., Williams CKI. Multi-task Gaussian Process Prediction (2007).
11. Wu J, Toscano-Palmerin S, Frazier PI, Wilson AG. Practical Multi-fidelity Bayesian Optimization for Hyperparameter Tuning. *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*.
12. Mainini L, Serani A, Rumpfkeil MP *et al.* (2022). *Analytical Benchmark Problems for Multifidelity Optimization Methods*.
13. Eriksson D, Pearce M, Gardner JR, Turner R, Poloczek M. Scalable Global Optimization via Local Bayesian Optimization doi:10.48550/arXiv.1910.01739 (2019) (Epub ahead of print).