Robust Multi-fidelity Bayesian Optimization with Deep Kernel and Partition

Fengxue Zhang
Department of Computer Science,
University of Chicago,
Illinois, U.S.

Thomas A. Desautels
Lawrence Livermore
National Laboratory,
California, U.S.

Yuxin Chen
Department of Computer Science,
University of Chicago,
Illinois, U.S.

Abstract

Multi-fidelity Bayesian optimization (MFBO) is a powerful approach that utilizes low-fidelity, cost-effective sources to expedite the exploration and exploitation of a high-fidelity objective function. Existing MFBO methods with theoretical foundations either lack justification for performance improvements over single-fidelity optimization or rely on strong assumptions about the relationships between fidelity sources to construct surrogate models and direct queries to low-fidelity sources. To mitigate the dependency on cross-fidelity assumptions while maintaining the advantages of low-fidelity queries, we introduce a random sampling and partition-based MFBO framework with deep kernel learning. This framework is robust to cross-fidelity model misspecification and explicitly illustrates the benefits of low-fidelity queries. Our results demonstrate that the proposed algorithm effectively manages complex cross-fidelity relationships and efficiently optimizes the target fidelity function.

1 Introduction

Multi-fidelity Bayesian optimization (MFBO) (Dai et al., 2019; Wu et al., 2020b; Takeno et al., 2020) is increasingly prevalent in the adaptive design of scientific experiments (Buterez et al., 2023), automated hyperparameter optimization (Eggensperger et al., 2021; Pfisterer et al., 2022), and policy optimization in control problems (Letham and Bakshy, 2019; Wu et al., 2020b).

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

Previous work has often relied on various assumptions about the relationship between different fidelities to analyze efficiency theoretically (Song et al., 2019; Kandasamy et al., 2016, 2017). Similar to transfer learning for Bayesian optimization (BO), a more practical challenge involves handling significant misalignment between fidelities while maintaining cost efficiency. Recent approaches to robust transfer learning for BO (Appice et al., 2015; Probst et al., 2019; Perrone et al., 2019; Reif et al., 2012; Pfisterer et al., 2021; Feurer et al., 2018) and robust single-fidelity BO against model misspecification (Bogunovic and Krause, 2021; Liu et al., 2023) have addressed this issue yet typically do not consider the sample efficiency on the lower fidelities. Some research suggests mitigating the problem by avoiding evaluations or learning from unreliable low-fidelity sources (Mikkola et al., 2023; Foumani et al., 2023), but they do not explicitly deal with errors incurred from the unreliable model learning of the multi-fidelity structure in the model design and acquisition.

Leveraging recent advancements in efficient kernel learning, uncertainty quantification, and error bounds for learning algorithms (Xu and Raginsky, 2017; Robinson et al., 2020; Wang et al., 2021), we propose a general-purpose framework that uses sampling-based cost-aware acquisition. This framework captures complex and potentially misaligned multi-fidelity evaluations while explicitly addressing model misspecification on the fly with robust data acquisition and deep kernel learning.

Our **key insight** is that due to kernel learning and the challenges in accurately specifying the prior, the surrogate model often underestimates the epistemic uncertainty (Yao et al., 2024). Though the problem of inaccurate surrogate models is universal in BO (Foumani et al., 2023; Liu et al., 2023), it is specific to MFBO that we could explicitly *improve the surrogate model in a cost-efficient manner* by querying the low-fidelity and low-cost sources. This insight differentiates our work from existing robust BO methods (Bogunovic

and Krause, 2021; Mikkola et al., 2023) that do not demonstrate the benefits of low-fidelity queries from the perspective of improving the model.

Our contributions are as follows:

- We propose a novel MFBO framework that features robustness to model misspecification through deep kernel learning and random sampling on low fidelities, together with constraining the target-fidelity query via partition of the search space. This novel framework allows us to explicitly depict the benefits of low-fidelity queries.
- We bound the proposed algorithm's regret under conventional assumptions on the underlying objective and additional assumptions based on recent advancements in multi-task learning.
- We demonstrate the effectiveness of the proposed algorithm in handling complex cross-fidelity relationships and efficient optimization of the target fidelity function.

2 Related Work

Overview of multi-fidelity BO Various approaches have been explored in multi-fidelity Bayesian optimization (MFBO), including methods with theoretical justifications (Song et al., 2019; Kandasamy et al., 2016, 2017) and practical-oriented methods that incorporate specific structures of multiple fidelities. Examples of the latter include early stopping (Dai et al., 2019), trace-aware MFBO (Wu et al., 2020b), and an entropy-based method for asynchronous parallel MFBO (Takeno et al., 2020).

The choice of the surrogate model plays an important role in MFBO algorithms. The common design choices include (1) learning one single model (Letham and Bakshy, 2019; Wu et al., 2020b), (2) learning multiple separately trained models combined through (weighted) addition (Feurer et al., 2018; Song et al., 2019); (3) learning neural network to capture the hierarchical structures (Li et al., 2020). However, there is no one-size-fits-all solution for handling different types of tasks since each design relies on specific problem structures. For example, early stopping in hyperparameter tuning and the independent impact on objectives between spatial knowledge and task-specific knowledge. HyperBand (Li et al., 2018), a closely related idea, mitigates the need to specify the parameter function by sampling multiple fidelities with manually defined budget allocation. However, it has limitations in handling complex multi-fidelity correlations. The recent advancement, PriorBand (Mallik et al., 2024), tailored for deep learning hyperparameter tuning, combines HyperBand with πBO (Hvarfner et al., 2020), but inherits the limitations of HyperBand in dealing with the complex multi-fidelity correlations.

Robust multi-fidelity BO Several approaches have been proposed to enhance robustness in MFBO. These include the exclusion of unreliable sources using distances on the learned latent space (Foumani et al., 2023) and safeguarding efficiency through vanilla BO by excluding inefficient low-fidelity queries with manually specified thresholds (Mikkola et al., 2023). However, these methods do not explicitly address errors arising from unreliable model learning in the multi-fidelity structure during both model design and acquisition. Specifically, Foumani et al. (2023) offers no theoretical justification for the accuracy of the latent space, while Mikkola et al. (2023) does not show improvement over single-fidelity theoretical guarantees. Our work is distinct in that it explicitly accounts for model misspecification and leverages the benefits of querying biased low-fidelity from the perspective of improving the model.

BO with misspecified models Works on Bayesian optimization with misspecified models typically introduce additional assumptions (Bogunovic and Krause, 2021; Liu et al., 2023) about the relationship between the assumed function space and the true underlying function. Due to the complexity of low-fidelity sources, developing universally appropriate assumptions is challenging. Furthermore, these works do not account for different evaluation costs, leading to a lack of a principled trade-off between misspecifications and costs.

Robust transfer learning for BO Robust transfer learning is conceptually closely related to addressing the misalignment of different data sources in MFBO. Recent works have improved the sample efficiency of Bayesian optimization in the target domain through various techniques, such as search space reduction (Appice et al., 2015; Probst et al., 2019; Perrone et al., 2019), initial point selection based on knowledge from previous tasks (Reif et al., 2012; Pfisterer et al., 2021), and robust modeling in meta-learning for Bayesian optimization (Feurer et al., 2018). However, these approaches are not directly applicable to MFBO as they do not consider the cost of queries in data sources other than the primary objective.

3 Preliminary

We begin by introducing useful notation, mostly following previous work by Song et al. (2019), and formally stating the problem studied in this paper.

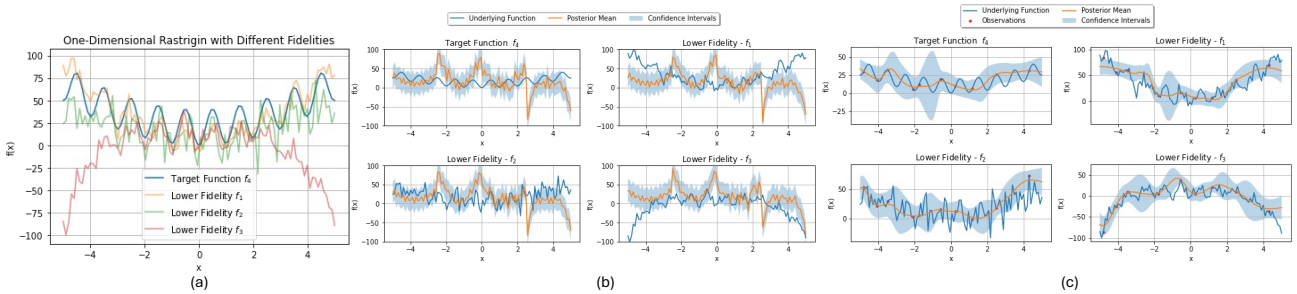

Figure 1: We illustrate the problem of learning and optimization on both target fidelity and misaligned low fidelities. (a) demonstrates the one-dimensional Rastrigin function (Pohlheim, 2007) and the manually constructed low fidelities. (b) demonstrates the posterior of learning the multi-fidelity functions with conventional multi-task GP (Swersky et al., 2013) previously applied in MFBO (Letham and Bakshy, 2019) when feeding 2000 training points densely distributed in the search space. (c) shows the posterior of the proposed model when feeding 10 points from each fidelity.

3.1 Multi-fidelity Optimization of Unknown Objective

Consider the problem of maximizing an unknown payoff function $f_M : \mathbf{X} \to \mathbb{R}$. We can probe this function by directly querying it at some point $\mathbf{x} \in \mathbf{X}$, consequently obtaining a noise-free observation $y_{\langle \mathbf{x}, M \rangle} = f_M(\mathbf{x})$.

In addition to f_M, we have access to oracle calls for unknown auxiliary functions $f_1, \ldots, f_{M-1} : \mathbf{X} \to \mathbb{R}$. Similarly, querying any f_ℓ at $\mathbf{x}$ yields a noise-free observation $y_t = f_{\ell_t}(\mathbf{x}_t)$.

Each auxiliary function f_ℓ can be viewed as a lower-fidelity version of f_M when $\ell < M$. Specifically, we model the unknown target fidelity functions with corresponding Gaussian process (GP): $f_M \sim \mathrm{GP}\left(\mu_M(\mathbf{x}), k_M(\mathbf{x}, \mathbf{x}')\right)$, where μ_M and k_M denote the prior mean and covariance.

Let $\langle \mathbf{x}, \ell \rangle$ denote the action of querying f_ℓ at $\mathbf{x}$. Each action $\langle \mathbf{x}, \ell \rangle$ incurs a cost of λ_ℓ and yields a reward:

$$r(\langle \mathbf{x}, \ell \rangle) = \begin{cases} f_M(\mathbf{x}) & \text{if } \ell = M \\ r_{\min} & \text{otherwise} \end{cases}$$

That is, performing $\langle \mathbf{x}, M \rangle$ at the target fidelity level achieves a reward $f_M(\mathbf{x})$. The collective historical observations after T iteration is denoted by $\mathcal{D}_T \triangleq \{\langle \mathbf{x}_t, \ell_t \rangle, y_t\}_{t=1\ldots T}$. We also define the collective historical observations up to certain fidelity ℓ as $\mathcal{D}_{\ell,T} \triangleq \{\langle \mathbf{x}_t, \ell'_t \rangle, y_t\}_{1 \le t \le T, \ell'_t \le \ell}$. When given a fixed budget, we need to guarantee the cumulative cost does not exceed the budget, i.e., $\sum_{t=1}^{T} \lambda_{\ell_t} \le \Lambda$.

Lower fidelity actions $\langle \mathbf{x}, \ell \rangle$ for $\ell < M$ yield the minimal immediate reward $r_{\min}$ but can provide valuable information about f_M, potentially leading to better decisions later. Without loss of generality, we assume $\max_{\mathbf{x}} f_M(\mathbf{x}) \ge 0$ and $r_{\min} \equiv 0$. Note that we define the reward only as incurred based on the target fidelity, and the query on low fidelity does not incur a reward but only helps with the learning. Hence, in the context of multi-fidelity Bayesian optimization, the **simple regret (SR)** is defined as: $\mathbf{R}(\hat{\mathbf{x}}) = f_M(\mathbf{x}^*) - f_M(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}} := \arg\max_{\mathbf{x}:(\langle \mathbf{x}, M \rangle, y) \in \mathcal{D}_T} f_M(\mathbf{x})$ is a point selected to be evaluated at the target fidelity, and $\mathbf{x}^*$ is the global maximizer of the function f_M. Our objective is to find the candidate that minimizes the simple regret after **exhausting a given budget** Λ.

3.2 Expected Excess Risk

In addition to the conventional analysis that assumes the prior is properly specified, we explicitly deal with the model misspecification regarding the difference between the posterior mean and true underlying function. In the context of statistical learning theory, the **convergence rate** of the expected excess risk with respect to the training dataset $\mathcal{D}_{\ell,T}$ at fidelity $\ell \in [M]^+ = [1 \ldots M]$ after T iterations is defined as $\mathrm{Rate}_{MF}(\ell, T) \triangleq \mathbb{E}\left[\mathcal{L}(f_\ell, \tilde{f}_\ell) | \mathcal{D}_{\ell,T}\right] - \mathcal{L}(f_\ell, \tilde{f}_\ell^*) = O(T^\alpha)$, Here, $-1 < \alpha < 0$ is a constant that characterizes the rate of convergence. $\tilde{f}_\ell$ is the hypothesis of fidelity ℓ produced by the learning algorithm when trained on $\mathcal{D}_{\ell,T}$. $\mathcal{L}(f, \tilde{f})$ is the loss function evaluating the hypothesis $\tilde{f}$ with respect to the true objective f, and $\tilde{f}_\ell^*$ is the hypothesis on fidelity ℓ that minimizes the expected loss.

3.3 Assumptions

To facilitate later discussion of the algorithm design and analysis, we first state the typical assumption for BO performance analysis.

Assumption 1 *Throughout the optimization,* f_M

bears an upper bound B on the RKHS norm (Salgia et al., 2024) corresponding to the learned deep kernels k_t. Namely $\|f_M\|_{\mathcal{H}_{k_t}} \leq B$.

Assumption 2 *(Generalized assumption 4.1 of Salgia et al. (2024)) For all $n \in \mathbb{N}$, there exists a discretization $\mathcal{S}_n$ of $\mathcal{X}$ such that for all $f_M \in \mathcal{H}_{k_t}$,*

$$|f_M(\mathbf{x}) - f_M([\mathbf{x}]_{\mathcal{S}_n})| \leq \|f_M\|_{\mathcal{H}_k}/n \text{ and } |\mathcal{S}_n| = poly(n),$$

where $[x]_{\mathcal{S}_n} = \arg\min_{y \in \mathcal{S}_n} \|x - y\|_2$, is the point in $\mathcal{S}_n$ that is closest to x.

Assumption 3 *(Assumption 4.2 of Salgia et al. (2024)) Let $\mathcal{L}_\eta = \{x \in \mathcal{X} \mid f_M(x) \geq \eta\}$ denote the level set of f_M for $\eta \in [-B, B]$. We assume that for all $\eta \in [-B, B]$, $\mathcal{L}_\eta$ is a disjoint union of at most $Q_{f_M} < \infty$ components, each of which is closed and connected. Moreover, for each such component, there exists a bi-Lipschitzian map[1] between each such component and $\mathcal{X}$ with normalized Lipschitz constant pair $L_{f_M}, L'_{f_M} < \infty$.*

Also, we state the assumption for analysis of the statistical learning.

Assumption 4 *The hypothesis space for each fidelity $\mathcal{H}_\ell$ contains the underlying functions f_ℓ for $\forall \ell \in [M]^+$.*

Assumption 5 *We assume that for fidelities $\ell \in \{2, \ldots, M\}$, the learned function $g_\ell \circ h$ for fidelity ℓ is L-Lipschitz relative to the function space $\mathcal{H}_h$. Formally, for all $\mathbf{x} \in \mathbf{X}$, $y \in \mathbf{Y}_\ell$, $h, h' \in \mathcal{H}_h$, the following inequality holds: $|\mathcal{L}(y, g_\ell(h(\mathbf{x}))) - \mathcal{L}(y, g_\ell(h'(\mathbf{x})))| \leq L\mathcal{L}(g_{\ell-1}(h(\mathbf{x})), g_{\ell-1}(h'(\mathbf{x})))$.*

This generalizes the assumption of Definition 4 from Robinson et al. (2020), that the lower fidelity performance perturbation could bound the higher fidelity performance change.

4 Method

In this section, we discuss the model design and the analysis-inspired data acquisition procedure of the proposed **R**obust **M**ulti-**F**idelity **B**ayesian **O**ptimization with **D**eep Kernel Learning and **P**artition (RMFBO-DP). The primary challenge in optimization-oriented multi-source model learning is the potential misalignment between sources, particularly when there is a lack of training data for model calibration due to the budget-sensitive nature of the optimization task. An efficient MFBO algorithm necessitates careful calibration of the model along with cost-effective optimization of the target fidelity.

We address these challenges in the model design and acquisition procedure of RMFBO-DP. From the perspective of model design, we employ a regularized deep kernel to capture the intricate multi-fidelity relationships. To improve the acquisition procedure, we first decouple the optimization objective into two parts: minimizing the target fidelity regret and improving the accuracy of the model itself. Then, we utilize random sampling on the low-fidelity sources to enhance the cost-efficiency of model calibration while constraining target-fidelity sampling to promising partitions by considering both optimality and potential model misspecification. We unfold the details of the model and acquisition procedure in the following subsections.

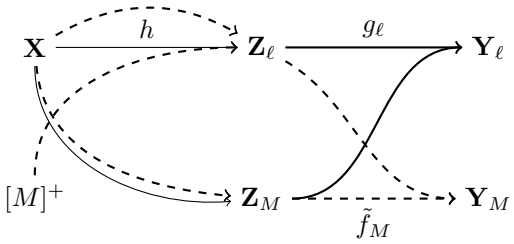

Figure 2: Schema for multi-fidelity learning implemented with deep kernel. The dotted lines denote the flow of target fidelity (strong data), and the solid lines the flow of low fidelities(weak data). Here, we denote the output space for certain fidelity $\ell \in [M]^+$ as $\mathbf{Y}_\ell$. Specifically, the target fidelity output space is denoted as $\mathbf{Y}_M$.

4.1 Model

We employ hierarchical deep kernel learning (Wilson et al., 2016) regularized with the spectral norm (Van Amersfoort et al., 2021) with mean absolute error (MAE) as the loss function to deal with overfitting caused by shortage of training data and the specific choice of training loss. We assume that the multiple fidelities $\{f_\ell\}_{\ell \in [M]}$ are mutually dependent through some fixed, possibly unknown joint probability distribution. Therefore, we seek to approximate the joint underlying function $f : \mathbf{X} \times [M]^+ \to \mathbb{R}$, where both the position $\mathbf{x} \in \mathbf{X}$ and fidelity $\ell \in [M]^+$ are inputs, and learn the approximation $\tilde{f}$ through joint learning of h and $g_{\ell \in [M]^+}$. Here, a *single* latent space mapping $h : \mathbf{X} \times [M]^+ \to \mathbf{Z}$ convert the input space $\mathbf{X}$ to the latent space $\mathbf{Z}$ which consists of the fidelity independent part $\mathbf{Z}^M$ and fidelity dependent part $\mathbf{Z}^\ell$. On top of the latent space, we construct a set of objective mappings $g_1 \ldots g_M$ for each fidelity. Namely

[1] A map $f : X \to Y$ is bi-Lipschitzian if there exist constants $c_1, c_2 > 0$ such that $c_1\|x - y\| \leq \|f_M(x) - f_M(y)\| \leq c_2\|x - y\|$ for all $x, y \in X$.

$\forall \ell \in [M]^+$, $f_\ell \triangleq f(\cdot, \ell)$ and $f(\cdot, \ell)$ is approximated by $\tilde{f}_\ell \triangleq g_\ell(h(\cdot, \ell))$. We illustrate the model structure in figure 2.

4.2 Data Acquisition

Random sampling within partitions We extend random exploration to the multi-fidelity regime. We use both the expected generalization error and SR bounds to guide the cost-efficient acquisition. When the SR contributes more to the general regret, we conduct target fidelity acquisition. Previous works offer an upper bound for cumulative regret when applying random exploration on a region of interest (Salgia et al., 2024) defined as $\{x \in \mathbf{X} \mid \mathrm{UCB}_{f_M,t}(\mathbf{x}) > \max_{\mathbf{x}' \in \mathbf{X}} \mathrm{LCB}_{f_M,t}(\mathbf{x}')\}$, as the target fidelity acquisition function when we ignore the contribution of low-fidelity evaluation to the target fidelity learning. We extend the cumulative regret bound into the following form of SR bound.

Theorem 1 *When assumptions 1, 2, and 3 hold, constraining the random exploration on the target fidelity on* $\{x \in \mathbf{X} \mid \mathrm{UCB}_{f_M,t}(\mathbf{x}) > \max_{\mathbf{x}' \in \mathbf{X}} LCB_{f_M,t}(\mathbf{x}')\}$, *and choosing* $\beta = B^2$, *we have the following bound with probability at least* $1 - \delta$,

$$SR(t) = \tilde{\mathcal{O}}\left(\sqrt{\frac{\gamma_{T_M(t)}}{T_M(t)}} \log \frac{T_M(t)}{\delta}\right) \tag{1}$$

Here $\tilde{\mathcal{O}}$ *means up to the logarithmic factor, and* $T_\ell(t) \triangleq |\{\langle \mathbf{x}_{t'}, \ell_{t'} \rangle, y_{t'}\}_{1 \le t' \le t, \ell_{t'} = \ell}|$ *denotes evaluations at fidelity* ℓ *among* t *evaluations.*

Misspecification-aware simple regret We exploit recent advancements in expected excess risk in meta-learning (Robinson et al., 2020) to extend the previous SR results to multiple weak learning sources. We state the theoretical results here while deferring the proof and other details to the Appendix A. First, we decompose the ultimate SR into separate components for the conventional regret and the generalization error.

Theorem 2 *Under the assumptions of Theorem 1 except for constraining random exploration in* $\hat{\mathbf{X}}_t$ *as defined in equation 2. The misspecification-aware Bayesian simple regret (*$SR_{MA} = f_M(\mathbf{x}^*) - \tilde{f}_M(\hat{\mathbf{x}})$*) of the proposed algorithm can be decomposed into the standard Bayesian simple regret (SR) and the rate term as follows:*

$$SR_{MA}(t) \le SR(t) + Rate_{MF}(M, t).$$

Expected excess risk in multi-fidelity deep kernel learning In the following, we generalize the meta-learning expected excess risk (Robinson et al., 2020; Xu and Raginsky, 2017) to multi-fidelity learning.

Theorem 3 *(Generalized theorem 10 of Robinson et al. (2020)) With the aforementioned assumptions 1, 2, 3, 4, and 5 hold, and the lowest single fidelity bears the convergence rate* $Rate_{MF}(1, T) = O(\sqrt{\frac{1}{T_1}})$, *The excessive risk bears the bound* $Rate_{MF}(\ell, t) \le$

$$O(Rate_{MF}(\ell - 1, t) + \frac{\sqrt{\left(\log_{T_\ell(t)} Rate_{MF}(\ell - 1, t) + 1\right) \log T_\ell(t)}}{T_\ell(t)})$$

Cost-efficient convergence strategies This allows us to differentiate the multiple fidelities' contribution to the target fidelity learning and regret minimization. Specifically, a cost-aware multi-fidelity acquisition could be made by minimizing the $\mathrm{SR}_{\mathrm{MA}}(T)$ such that the total cost incurred by querying different fidelities does not exceed Λ. When the error bound contributes more to global regret, we randomly explore until it is more cost-efficient to conduct target fidelity acquisition. The detailed procedure is summarized in algorithm 1. We continue to discuss the key components of the acquisition procedure.

4.3 Additional Design Choices

Reliable search space exclusion We leverage the GP posterior calibrated with the excess risk on observed points to exclude from acquisition the regions that, with high probability, do not contain the global optimum. To do so, we rely on both the upper confidence bound $\mathrm{UCB}_{f_M,t}(\mathbf{x}) \triangleq \mu_{f_M,t-1}(\mathbf{x}) + \beta^{1/2}_{f_M,t}\sigma_{f_M,t-1}(\mathbf{x})$ and lower confidence bound $\mathrm{LCB}_{f_M,t}(\mathbf{x}) \triangleq \mu_{f_M,t-1}(\mathbf{x}) - \beta_t^{1/2}\sigma_{f_M,t-1}(\mathbf{x})$, where β_t is the scaling factor corresponding to certain confidence. Formally, the acting search space at iteration t is $\hat{\mathbf{X}}_t$ defined as

$$\left\{x \in \mathbf{X} \mid \mathrm{UCB}_{f_M,t}(\mathbf{x}) > \max_{\mathbf{x}' \in \mathbf{X}} \mathrm{LCB}'_{f_M,t}(\mathbf{x}')\right\} \tag{2}$$

Here $\mathrm{LCB}'_{f_M,t}(\mathbf{x}') \triangleq \mathrm{LCB}_{f_M,t}(\mathbf{x}') - \mathrm{Rate}_{MF}(M, t)$ generalize the lower confidence bound to incorporate the expected generalization error.

Estimation of SR rates Due to the difficulty of analyzing exact SR, we rely on the following simple approximation. For two consecutive evaluations of the target fidelity, if we observe improvement in the best reward, we leverage the improvement to regress the SR. Namely, for $\forall 1 \le t_1 < t_2 \le T$, if $\Delta_{f_M} \triangleq y_{\langle \mathbf{x}_{t_2}, \ell_t = M \rangle} - y_{\langle \mathbf{x}_{t_1}, \ell_t = M \rangle} > 0$, we update the approximation for $\mathrm{SR}(t_2)$ by solving $\Delta_{f_M} = \mathrm{SR}(t_2) - \mathrm{SR}(t_1)$.

Algorithm 1 **R**obust **M**ulti-**F**idelity **B**ayesian **O**ptimization with **D**eep Kernel Learning and **P**artition (RMFBO-DP)

Input:Search space $\mathbf{X}$, t_1 initial observation $\mathcal{D}_{t_1}$, total budget Λ;
Initialize timestamp as $t \leftarrow t_1$.
Initialize remaining budget $\Lambda \leftarrow \Lambda - \sum_{t'=1}^{t} \lambda_{\ell_{t'}}$
while $\Lambda > 0$ **do**
  Update the model $\tilde{f}$ as discussed in section 4.1.
  Identify ROIs $\hat{\mathbf{X}}_t$ according to equation 2.
  Maximize cost-sensitive $\mathrm{SR}_{\mathrm{MA}}(t+1) - \mathrm{SR}_{\mathrm{MA}}(t)$ reduction as discussed in section 4.2 Theorem 2.
  $\Delta_{\mathrm{SR},t} \leftarrow \frac{\mathrm{SR}(t+1)-\mathrm{SR}(t)}{\lambda_M}$
  $\Delta_{\mathrm{Rate}_{MF},t} \leftarrow \max_{\ell \in [M]^+} \frac{\mathrm{Rate}_{MF}(\ell,t+1)-\mathrm{Rate}_{MF}(\ell,t)}{\lambda_\ell}$
  if $\Delta_{\mathrm{SR},t} \geq \Delta_{\mathrm{Rate}_{MF},t}$ **then**
    Sample $\mathbf{x}_t$ from $\hat{\mathbf{X}}_t$ on target fidelity M
    $\mathbf{x}_t \leftarrow$ sample from uniform distribution on $\hat{\mathbf{X}}_t$.
  else
    Sample candidate on on fidelity ℓ_t
    $\ell_t \leftarrow \arg\max_{\ell \in [M]^+} \frac{\mathrm{Rate}_{MF}(\ell,t+1)-\mathrm{Rate}_{MF}(\ell,t)}{\lambda_\ell}$
    $\mathbf{x}_t \leftarrow$ sample from uniform distribution on $\mathbf{X}$.
  end if
  Update the observation $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{\langle \mathbf{x}_t, \ell_t' \rangle, y_t\}$
  Update the remaining budget $\Lambda \leftarrow \Lambda - \lambda_\ell$
  Update the timestamp $t \leftarrow t+1$
end while

Estimation of expected excess risk Similar to the above approximation of SR, we approximate the excess risk reduction by regressing to the observed fitting error improvement. $\mathrm{Rate}_{MF}(M,t)$. for $\forall 1 \leq t_1 < t_2 \leq T, \ell \in [M]^+$, we resort to 5-fold cross-validation on $\mathcal{D}_{\ell,t_1}$ and $\mathcal{D}_{\ell,t_2}$ to estimate the model fitting improvement $\Delta_{\mathcal{L}(f_\ell,\tilde{f}_\ell),t} = \mathrm{Rate}_{MF}(\ell,t_2) - \mathrm{Rate}_{MF}(\ell,t_1)$. Solving the equation allows us to approximate $\mathrm{Rate}_{MF}(\ell,t_2)$.

Constraining acquisition We rely on random discretization to constrain the acquisition within $\hat{\mathbf{X}}_t$, which rejects the candidates outside $\hat{\mathbf{X}}_t$. Note popular BO frameworks typically allow optimizing the acquisition function subject to constraints, e.g., Botorch (Balandat et al., 2020).

5 Experiments

We compare the proposed RMFBO-DP against four baselines. The first one is the entropy-based method denoted as MF-MES proposed by Takeno et al. (2020); the second one is denoted as MF-KG, which is the cost-efficient knowledge gradient method proposed by Wu et al. (2020b). The third and fourth algorithms are corresponding variants when applying the robust MFBO framework proposed by Mikkola et al. (2023), denoted as rMF-MES and rMF-KG correspondingly. The baselines chosen represent the latest developments in MFBO and corresponding robust variants. We rely on BoTorch (Balandat et al., 2020) and gpytorch (Gardner et al., 2018) to implement RMFBO-DP and the baselines. More baselines and comparisons are available in Appendix C.

5.1 Dataset

We evaluate algorithm performance on synthetic datasets and real-world tasks, including multi-fidelity protein design and hyperparameter optimizations.

Rastrigin Dataset As illustrated in figure 1, we construct the four-fidelity version of Rastrigin function (Pohlheim, 2007) on 1D search space. We further extend the construction to 20D search space. Here, the first two lower fidelities generally share the same trend as the target-fidelity underlying function, while the lowest fidelity that incurs the cheapest evaluation cost disagrees with the target fidelity function except for limited central area and largely diverges in the border areas.

Multi-fidelity Protein Design We use a protein engineering dataset describing a set of antigen/antibody binding calculations. These calculations, executed using supercomputing resources, estimate the change in binding free energy at the interface between each of the 71769 modified antibodies and the SARS-CoV-2 spike protein, as compared to the single reference antibody from which they are derived. Estimations of binding free energy ($\Delta\Delta G$) are calculated using protein-structure-based STATIUM (DeBartolo et al., 2014), Rosetta Flex simulation software (Das and Baker, 2008; Barlow et al., 2018) and FoldX (Schymkowitz et al., 2005; Sapozhnikov et al., 2023; Buß et al., 2018). We treat Rosetta's outcomes as the objective of the target fidelity. These calculations took several CPU hours each and were produced during an antibody design process (Desautels et al., 2020, 2022).

Yahoo GYM Introduced by Pfisterer et al. (2022), the Yahoo GYM dataset is crafted to address the shortcomings of traditional tabular benchmarks by offering a more nuanced evaluation environment. It incorporates varying levels of data fidelity, simulating real-world evaluation with surrogate models pre-trained on corresponding tasks. Yahoo GYM consists of three Hyperparameter Optimization tasks that were tested in the experiments and show that the surrogate-model-based evaluation offers reliable comparisons of different algorithms. It includes the following three different benchmarks.

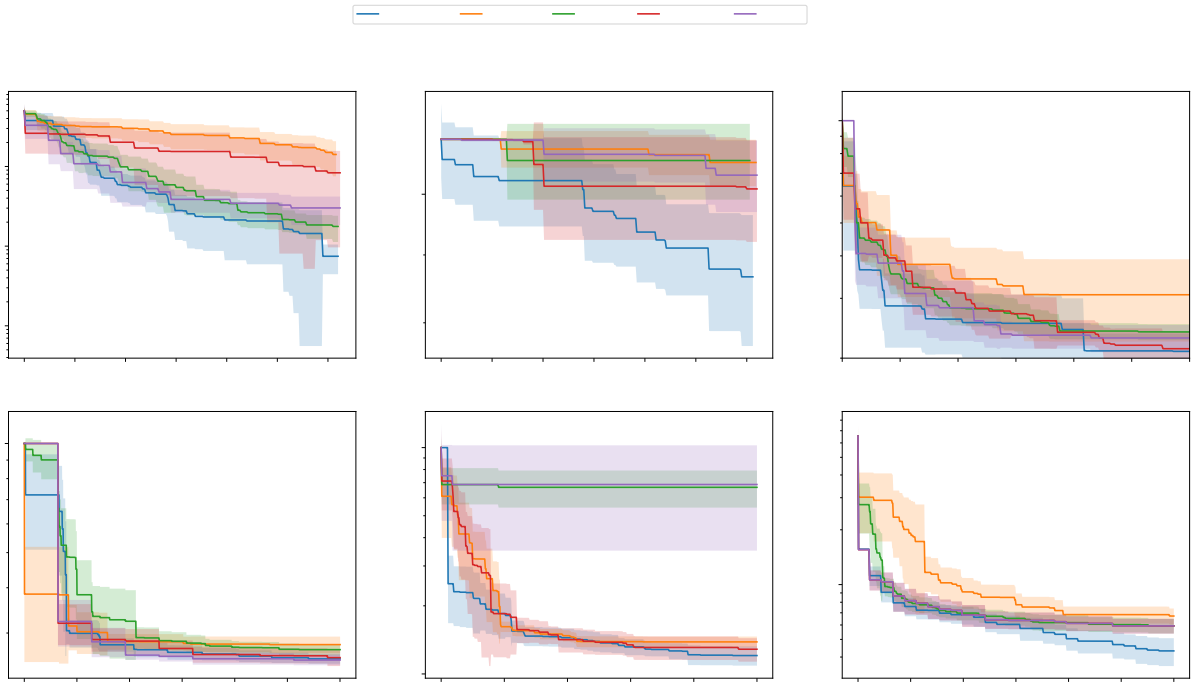

Figure 3: We illustrate the performance of RMFBO-DP compared against MF-MES, MF-KG, and Random algorithms on both synthetic and real-world datasets. The results are collected from at least ten independent trials. The y-axis denotes the simple regret, and the x-axis denotes the consumed budget. The shades area shows the 95% confidence interval. We trim the shared initial single data point.

RPART and XGBoost Two classification tasks optimized for the area under the curve (AUC) after five-fold cross-validation via mlr3pipelines (Binder et al., 2021). Fidelity is defined by the fraction used in the training set. RPART (Therneau et al., 1997) has 5 numerical hyperparameters; XGBoost (Chen and Guestrin, 2016) has 13.

LCBench A benchmark for studying multi-fidelity and meta-learning in deep learning (DL) (Zimmer et al., 2021), including learning curves for 2000 configurations across 35 datasets. Fidelity is specified by the epoch number, and the primary objective is balanced test set accuracy.

More detailed discussions about the experiment setup are deferred to Appendix B.

5.2 Evaluation

We evaluated the proposed algorithm Robust Multi-fidelity Bayesian Optimization (RMFBO-DP) against four baselines on both synthetic datasets corresponding to figure 1 and four real-world tasks. We show that the proposed algorithm outperforms the baselines in terms of simple regret on Rastrigin-1D, Rastrgin 20d, and Protein-86D. We also demonstrate that RMFBO-DP performance consistently matches the best of the baselines on Hyperparameter optimization tasks on Yahoo GYM, including RPART-4D, LCBench-7D, and XGBoost-13D.

Note that the performance of the baselines is inconsistent across different tasks we have tested. For example, MF-KG and rMF-KG outperform MF-MES and rMF-MES on Rastrigin-1D and Protein-86D, while the performance of rMF-MES is better on RPART-4D, LCBench, and XGBoost-13D. We found that on XGBoost-13D, the performance of rMF-KG and MF-KG is far behind the other algorithms as it is trapped in low-fidelity sources for most of iterations through the optimization. A similar problem occurs on RPART-4D, where the performance of MF-MES is behind the other algorithms as it is occasionally trapped in the low fidelities sources. Though rMF-MES shows improved performance over the base algorithm MF-MES in some scenarios, the previous robust MFBO framework proposed by Mikkola et al. (2023) framework fails to improve the MF-KG on Rastrigin-1D, Rastrigin-20D, RPART-4D, XGBoost-13D Protein-86D in our experiments. We hypothesize that Mikkola et al. (2023) does not explicitly address the problem of low-fidelity sources trapping the algorithm. In contrast, our pro-

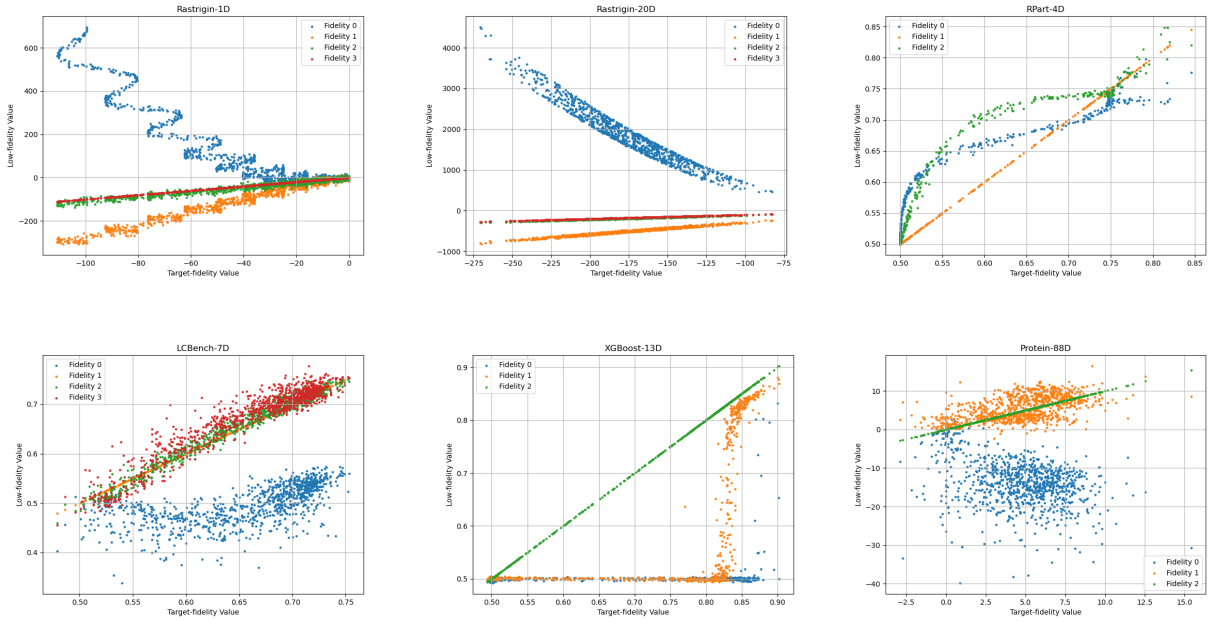

Figure 4: We illustrate the distribution of the low-fidelity sources and the target fidelity function on the benchmark tasks. The x-axis denotes the target-fidelity value, and the y-axis denotes the low-fidelity value. The scatter points are collected from the evaluation of the low-fidelity sources and the target fidelity function of 1000 random samples.

posed algorithm RMFBO-DP's relies on the estimation of SR improvements for budget allocation across different fidelities and demonstrates more consistent performance.

To better understand the performance difference of RMFBO-DP on Yahoo GYM, which fails to outperform the baselines, we further investigate the relationship between the low-fidelity sources and the target fidelity on different tasks. As is shown in figure 4, we found that the low-fidelity sources in Yahoo GYM are less biased, especially since the optimality does not contradict the target fidelity. While the trend of target fidelity function is largely different from the lowest fidelity sources for Rastrigin and Protein design task. This observation suggests that it is likely the advantage of RMFBO-DP is more outstanding when the cross-fidelity relationship is complex but learnable. In other cases, RMFBO-DP is less favorable as its unconstrained low-fidelity sampling does not optimize the low-fidelity sources as conventional MFBO algorithms.

5.3 Ablation Study

We conduct an ablation study to investigate the impact of the proposed algorithm components. We compare the performance of the additional variants of the baselines when applying the same deep kernel learning yet without random sampling on low-fidelities. As is shown in table 1, the performance of the variants is not consistently improved upon the corresponding baselines and lags behind RMFBO-DP. This observation suggests that data acquisition is crucial to the performance improvement of the proposed RMFBO-DP.

Method	Rastrigin-1D	Rastrigin-20D
RMFBO-DP	**0.75 ± 0.30**	**86.64 ± 9.80**
MF-MES	15.00 ± 6.83	105.66 ± 4.25
MF-KG	1.84 ± 0.61	106.03 ± 6.96
MF-MES-DK	2.83 ± 2.57	106.62 ± 5.72
MF-KG-DK	7.04 ± 4.60	110.00 ± 0.00
rMF-MES	8.33 ± 7.36	100.93 ± 8.87
rMF-KG	3.02 ± 1.18	103.37 ± 6.41
rMF-MES-DK	14.88 ± 7.65	103.20 ± 6.24
rMF-KG-DK	3.37 ± 1.23	103.52 ± 5.32
Random	5.05 ± 2.43	104.06 ± 6.15

Table 1: Ultimate simple regrets for Rastrigin-1D and Rastrigin-20D tasks. The best-performing algorithm for each task is highlighted in bold. The results are collected from at least ten independent trials. We mark the variants using a deep kernel with -DK.

6 Discussion

In this paper, we introduced a novel multi-fidelity Bayesian optimization (MFBO) approach focusing on

robustness and efficiency. Our method explicitly addresses the misspecification issues in multi-fidelity deep model learning by incorporating budget-sensitive low-fidelity sampling and constraining acquisitions to a subset of the global search space for target fidelity optimization. By tackling the challenges of low-fidelity misalignment and efficient target fidelity optimization in a principled, cost-effective manner, we demonstrated that our approach significantly improves robustness and performance over existing methods, as confirmed by our theoretical and empirical results.

Future work will explore different cost-efficient strategies, such as cost-efficient Expected Improvement (Snoek et al., 2012; Astudillo et al., 2021), which is known to be one-step Bayesian optimal, and cost-sensitive max-value entropy methods that enable efficient parallel multi-fidelity queries (Takeno et al., 2020). We believe our contributions, which introduce recent advancements in statistical learning theory within multi-task frameworks, represent a significant step towards a more robust MFBO. These advancements offer new insights and promising directions for future research in the field.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-CONF-872542. The GUIDE program is executed by the Joint Program Executive Office for Chemical, Biological, Radiological and Nuclear Defense (JPEO-CBRND) Joint Project Lead for CBRND Enabling Biotechnologies (JPL CBRND EB) on behalf of the Department of Defense's Chemical and Biological Defense Program. This effort was in collaboration with the Defense Health Agency (DHA) COVID funding initiative. The views expressed in this paper reflect the views of the authors and do not necessarily reflect the position of the Department of the Army, Department of Defense, nor the United States Government. References to non-federal entities do not constitute nor imply Department of Defense or Army endorsement of any company or organization.

References

Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, Carlos Soares, João Gama, and Alípio Jorge. Machine learning and knowledge discovery in databases. In *Proceedings of the European Conference, ECML PKDD, Part II, Porto, Portugal.–2015*. Springer, 2015. 1, 2

Raul Astudillo, Daniel R Jiang, Maximilian Balandat, Eytan Bakshy, and Peter I Frazier. Multi-step budgeted bayesian optimization with unknown evaluation costs: Supplementary material. In *35th Conference on Neural Information Processing Systems*, 2021. 9

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020. 6

Kyle A Barlow, Shane O Conchuir, Samuel Thompson, Pooja Suresh, James E Lucas, Markus Heinonen, and Tanja Kortemme. Flex ddg: Rosetta ensemble-based estimation of changes in protein–protein binding affinity upon mutation. *The Journal of Physical Chemistry B*, 122(21):5389–5399, 2018. 6, 20

Martin Binder, Florian Pfisterer, Michel Lang, Lennart Schneider, Lars Kotthoff, and Bernd Bischl. mlr3pipelines-flexible machine learning pipelines in r. *Journal of Machine Learning Research*, 22(184): 1–7, 2021. 7

Ilija Bogunovic and Andreas Krause. Misspecified gaussian process bandit optimization. *Advances in neural information processing systems*, 34:3004–3015, 2021. 1, 2

Oliver Buß, Jens Rudat, and Katrin Ochsenreither. Foldx as protein engineering tool: better than random based approaches? *Computational and structural biotechnology journal*, 16:25–33, 2018. 6, 20

David Buterez, Jon Paul Janet, Steven J Kiddle, and Pietro Liò. Mf-pcba: Multifidelity high-throughput screening benchmarks for drug discovery and machine learning. *Journal of Chemical Information and Modeling*, 63(9):2667–2678, 2023. 1

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785. 7

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017. 13

Zhongxiang Dai, Haibin Yu, Bryan Kian Hsiang Low, and Patrick Jaillet. Bayesian optimization meets bayesian optimal stopping. In *International conference on machine learning*, pages 1496–1506. PMLR, 2019. 1, 2

Rhiju Das and David Baker. Macromolecular modeling with rosetta. *Annu. Rev. Biochem.*, 77:363–382, 2008. 6, 20

Joe DeBartolo, Mikko Taipale, and Amy E Keating. Genome-wide prediction and validation of peptides that bind human prosurvival bcl-2 proteins. *PLoS computational biology*, 10(6):e1003693, 2014. 6, 20

Thomas Desautels, Adam Zemla, Edmond Lau, Magdalena Franco, and Daniel Faissol. Rapid in silico design of antibodies targeting sars-cov-2 using machine learning and supercomputing. *BioRxiv*, 2020. 6, 19

Thomas A Desautels, Kathryn T Arrildt, Adam T Zemla, Edmond Y Lau, Fangqiang Zhu, Dante Ricci, Stephanie Cronin, Seth Zost, Elad Binshtein, Suzanne M Scheaffer, et al. Computationally restoring the potency of a clinical antibody against sars-cov-2 omicron subvariants. *bioRxiv*, pages 2022–10, 2022. 6, 19

Katharina Eggensperger, Philipp Müller, Neeratyoy Mallik, Matthias Feurer, René Sass, Aaron Klein, Noor Awad, Marius Lindauer, and Frank Hutter. Hpobench: A collection of reproducible multi-fidelity benchmark problems for hpo. *arXiv preprint arXiv:2109.06716*, 2021. 1

Matthias Feurer, Benjamin Letham, Frank Hutter, and Eytan Bakshy. Practical transfer learning for bayesian optimization. *arXiv preprint arXiv:1802.02219*, 2018. 1, 2

Zahra Zanjani Foumani, Mehdi Shishehbor, Amin Yousefpour, and Ramin Bostanabad. Multi-fidelity cost-aware bayesian optimization. *Computer Methods in Applied Mechanics and Engineering*, 407:115937, 2023. 1, 2

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018. 6

Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. π bo: Augmenting acquisition functions with user beliefs for bayesian optimization. In *International Conference on Learning Representations*, 2020. 2

Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. *Advances in neural information processing systems*, 29, 2016. 1, 2

Kirthevasan Kandasamy, Gautam Dasarathy, Jeff Schneider, and Barnabás Póczos. Multi-fidelity bayesian optimisation with continuous approximations. In *International conference on machine learning*, pages 1799–1808. PMLR, 2017. 1, 2

Benjamin Letham and Eytan Bakshy. Bayesian optimization for policy search via online-offline experimentation. *Journal of Machine Learning Research*, 20(145):1–30, 2019. 1, 2, 3

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018. 2

Shibo Li, Wei Xing, Robert Kirby, and Shandian Zhe. Multi-fidelity bayesian optimization via deep neural networks. *Advances in Neural Information Processing Systems*, 33:8521–8531, 2020. 2

Chong Liu, Ming Yin, and Yu-Xiang Wang. No-regret linear bandits beyond realizability. In *Uncertainty in Artificial Intelligence*, pages 1294–1303. PMLR, 2023. 1, 2

Neeratyoy Mallik, Edward Bergman, Carl Hvarfner, Danny Stoll, Maciej Janowski, Marius Lindauer, Luigi Nardi, and Frank Hutter. Priorband: Practical hyperparameter optimization in the age of deep learning. *Advances in Neural Information Processing Systems*, 36, 2024. 2

Petrus Mikkola, Julien Martinelli, Louis Filstroff, and Samuel Kaski. Multi-fidelity bayesian optimization with unreliable information sources. In *International Conference on Artificial Intelligence and Statistics*, pages 7425–7454. PMLR, 2023. 1, 2, 6, 7, 20

Valerio Perrone, Huibin Shen, Matthias W Seeger, Cedric Archambeau, and Rodolphe Jenatton. Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. *Advances in neural information processing systems*, 32, 2019. 1, 2

Florian Pfisterer, Jan N van Rijn, Philipp Probst, Andreas C Müller, and Bernd Bischl. Learning multiple defaults for machine learning algorithms. In *Proceedings of the genetic and evolutionary computation conference companion*, pages 241–242, 2021. 1, 2

Florian Pfisterer, Lennart Schneider, Julia Moosbauer, Martin Binder, and Bernd Bischl. Yahpo gym-an efficient multi-objective multi-fidelity benchmark for hyperparameter optimization. In *International Conference on Automated Machine Learning*, pages 3–1. PMLR, 2022. 1, 6

Hartmut Pohlheim. Examples of objective functions. *Retrieved*, 4(10):2012, 2007. 3, 6

Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. Tunability: Importance of hyperparameters

of machine learning algorithms. *Journal of Machine Learning Research*, 20(53):1–32, 2019. 1, 2

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. 20

Matthias Reif, Faisal Shafait, and Andreas Dengel. Meta-learning for evolutionary parameter optimization of classifiers. *Machine learning*, 87:357–380, 2012. 1, 2

Joshua Robinson, Stefanie Jegelka, and Suvrit Sra. Strength from weakness: Fast learning using weak supervision. In *International Conference on Machine Learning*, pages 8127–8136. PMLR, 2020. 1, 4, 5, 15

Sudeep Salgia, Sattar Vakili, and Qing Zhao. Random exploration in bayesian optimization: Order-optimal regret and computational efficiency. In *Forty-first International Conference on Machine Learning*, 2024. 4, 5, 13, 14

Yesol Sapozhnikov, Jagdish Suresh Patel, F Marty Ytreberg, and Craig R Miller. Statistical modeling to quantify the uncertainty of foldx-predicted protein folding and binding stability. *BMC bioinformatics*, 24(1):426, 2023. 6, 20

Joost Schymkowitz, Jesper Borg, Francois Stricher, Robby Nys, Frederic Rousseau, and Luis Serrano. The foldx web server: an online force field. *Nucleic acids research*, 33(suppl_2):W382–W388, 2005. 6, 20

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *26th Annual Conference on Neural Information Processing Systems 2012*, pages 2951–2959, 2012. 9

Jialin Song, Yuxin Chen, and Yisong Yue. A general framework for multi-fidelity bayesian optimization with gaussian processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3158–3167. PMLR, 2019. 1, 2

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009. 13, 20

Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task bayesian optimization. *Advances in neural information processing systems*, 26, 2013. 3

Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity bayesian optimization with max-value entropy search and its parallelization. In *International Conference on Machine Learning*, pages 9334–9345. PMLR, 2020. 1, 2, 6, 9

Terry M Therneau, Elizabeth J Atkinson, et al. An introduction to recursive partitioning using the rpart routines. Technical report, Technical report Mayo Foundation, 1997. 7

Sattar Vakili, Nacime Bouziani, Sepehr Jalali, Alberto Bernacchia, and Da-shan Shiu. Optimal order simple regret for gaussian process bandits. *Advances in Neural Information Processing Systems*, 34:21202–21215, 2021. 13

Joost Van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021. 4, 20

Zifeng Wang, Shao-Lun Huang, Ercan E Kuruoglu, Jimeng Sun, Xi Chen, and Yefeng Zheng. Pac-bayes information bottleneck. *arXiv preprint arXiv:2109.14509*, 2021. 1

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In Arthur Gretton and Christian C. Robert, editors, *19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 370–378, Cadiz, Spain, 09–11 May 2016. PMLR. 4, 20

Jian Wu, Saul Toscano-Palmerin, Peter I. Frazier, and Andrew Gordon Wilson. Practical multi-fidelity bayesian optimization for hyperparameter tuning. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 788–798. PMLR, 22–25 Jul 2020a. URL https://proceedings.mlr.press/v115/wu20a.html. 20

Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. Practical multi-fidelity bayesian optimization for hyperparameter tuning. In *Uncertainty in Artificial Intelligence*, pages 788–798. PMLR, 2020b. 1, 2, 6

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in neural information processing systems*, 30, 2017. 1, 5

Michael S Yao, Yimeng Zeng, Hamsa Bastani, Jacob Gardner, James C Gee, and Osbert Bastani. Generative adversarial bayesian optimization for surrogate objectives. *arXiv preprint arXiv:2402.06532*, 2024. 1

Lucas Zimmer, Marius Lindauer, and Frank Hutter. Auto-pytorch: Multi-fidelity metalearning for efficient and robust autodl. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3079–3090, 2021. 7

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes] We will include the proof in supplementary material.
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] We will discuss the detailed setting in appendix.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable.[Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Proofs

Here we present the missing proofs for the theorems in the main text. We begin by introducing the necessary notions.

Definition 1 (RKHS (Salgia et al., 2024)) *Consider a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. A Hilbert space $\mathcal{H}_k$ of functions on $\mathcal{X}$ equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$ is called a Reproducing Kernel Hilbert Space (RKHS) with reproducing kernel k if the following conditions are satisfied: (i) $\forall x \in \mathcal{X}, k(\cdot, x) \in \mathcal{H}_k$; (ii) $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}_k, f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}_k}$. For simplicity, we use ψ_x to denote $k(\cdot, x)$. The inner product induces the RKHS norm, $\|f\|^2_{\mathcal{H}_k} = \langle f, f \rangle_{\mathcal{H}_k}$.*

We also generalize the maximum mutual information gain in the multi-fidelity optimization setting as follows.

Definition 2 (Maximum Information Gain (Srinivas et al., 2009)) *The information gain of a function f refers to the mutual information between the distribution of f and the distribution of t observations of the function, denoted as $I(Y_t; f)$. Let the set of instances corresponding to each element in Y_t be A_t.*

$$I(Y_t; f) = H(f) - H(f|Y_t) = H(Y_t) - H(Y_t|f)$$

where $H(\cdot)$ is the Shannon entropy function. The closed-form expression for $I(Y_t; f)$ is given by $\frac{1}{2} \log \det(I + \lambda^{-1} K_{i,t})$ where $K_t = [k(\mathbf{x}, \mathbf{x}')_{\mathbf{x}, \mathbf{x}' \in A_t}]$. Consequently, the maximum information gain (MIG) for the objective f given t observations is defined as:

$$\gamma_t = \max_{A_t \subset \mathbf{X} \ s.t. \ |A_t| = t} \frac{1}{2} \log \det(I + \sigma^{-2} K_t)$$

However, due to the kernel learning in the multi-fidelity optimization setting, the kernel associated with the assumptions is constantly changing. We extend the above definition to the multi-fidelity setting as follows.

Definition 3 (Maximum Information Gain in Multi-fidelity Optimization) *The maximum information gain regarding the target-fidelity objective distribution f_M is defined as follows.*

$$I(\mathcal{D}_{\ell,t}; f_M) = \frac{1}{2} \log \det(I + \lambda^{-1} K_{i,t'})$$

where $K_{t'} = [k_t(\mathbf{x}, \mathbf{x}')_{\mathbf{x}, \mathbf{x}' \in \mathcal{D}_{\ell,t}}]$. For simplicity, we use the notation $\gamma_t = \max_{\mathcal{D}_{\ell,t}, s.t. |\mathcal{D}_{\ell,t}| = t'} \frac{1}{2} \log \det(I + \sigma^{-2} K_{t'})$ in all the following discussions.

A.1 Proof of Theorem 1

We provide necessary lemmas and proofs to establish the theorem. There are two major differences between our results and the previous results in Salgia et al. (2024). First, we consider kernel learning in the multi-fidelity optimization setting, where the kernel associated with the assumptions is constantly changing. Second, we consider the noise-free setting, yet our results resemble the bound corresponding to the noisy observation scenario.

The first lemma justifies the definition of the identified region of interest. For simplicity, we denote $\{x \in \mathbf{X} \mid \mathrm{UCB}_{f_M,t}(\mathbf{x}) > \max_{\mathbf{x}' \in \mathbf{X}} \mathrm{LCB}_{f_M,t}(\mathbf{x}')\}$ as $\hat{\mathbf{X}}'_t$.

Lemma 1 *We denote $\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathbf{X}} f_M(\mathbf{x})$. Then $\forall t \geq 1$, we have $\mathbf{x}^* \in \hat{\mathbf{X}}'_t \subseteq \mathbf{X}$.*

Proof: Under Assumption 1, we obtain the following results as in Vakili et al. (2021); Chowdhury and Gopalan (2017) that $\forall t \geq 1$,

$$|f_M(\mathbf{x}) - \mu_{f_M}(\mathbf{x})| \leq B \sigma_{f_M, t-1}$$

Then we have the following inequalities which reflect the desired property of $\hat{\mathbf{X}}'_t$.

$$\max_{\mathbf{x} \in \mathbf{X}} \mathrm{LCB}_{f_M,t}(\mathbf{x}) \leq f_M(\mathbf{x}^*) \leq \max_{\mathbf{x} \in \mathbf{X}} \mathrm{UCB}_{f_M,t}(\mathbf{x})$$

□

The second lemma justifies the random selection within the identified region of interest.

Lemma 2 *$\forall t \geq 1$, we have $f_M(\mathbf{x}^*) - \min_{\mathbf{x}\in\hat{\mathbf{X}}'_t} f_M(\mathbf{x}) \leq 4B \max_{\mathbf{x}\in\hat{\mathbf{X}}'_t} \sigma_{f_M,t}$. Specifically, when $t = 1$, $f_M(\mathbf{x}^*) - \min_{\mathbf{x}\in\hat{\mathbf{X}}'_t} f_M(\mathbf{x}) \leq 2B$.*

Proof: When $t = 1$, we have

$$f_M(\mathbf{x}^*) - \min_{\mathbf{x}\in\hat{\mathbf{X}}'_t} f_M(\mathbf{x}) \leq 2f_M(\mathbf{x}^*) \leq 2B$$

given the assumptions.

When $t > 1$, we denote $\mathbf{x}' = \arg\min_{\mathbf{x}\in\hat{\mathbf{X}}'_t} f_M(\mathbf{x})$. Then we have

$$\mathrm{UCB}_{f_M,t}(\mathbf{x}') - \mathrm{LCB}_{f_M,t}(\mathbf{x}') \leq 2B \max_{\mathbf{x}\in\hat{\mathbf{X}}'_t} \sigma_{f_M,t}$$

At the same time, we have

$$\max_{\mathbf{x}\in\mathbf{X}} \mathrm{UCB}_{f_M,t}(\mathbf{x}) - \mathrm{UCB}_{f_M,t}(\mathbf{x}') \leq 2B \max_{\mathbf{x}\in\hat{\mathbf{X}}'_t} \sigma_{f_M,t}$$

given the definition of $\hat{\mathbf{X}}'_t$. Combining the above two inequalities, we have

$$f_M(\mathbf{x}^*) - \min_{\mathbf{x}\in\hat{\mathbf{X}}'_t} f_M(\mathbf{x}) \leq \max_{\mathbf{x}\in\mathbf{X}} \mathrm{UCB}_{f_M,t}(\mathbf{x}) - \mathrm{LCB}_{f_M,t}(\mathbf{x}') \leq 4B \max_{\mathbf{x}\in\hat{\mathbf{X}}'_t} \sigma_{f_M,t}$$

□

Then we can prove Theorem 1 as follows with Theorem 4.5 from Salgia et al. (2024).

Proof: With Lemma 1 and Lemma 2, we fulfill the conditions to apply Theorem 4.5 from Salgia et al. (2024) to the multi-fidelity optimization setting. Specifically, we have after sufficient iterations

$$\sum_{t=1}^{T} (f_M(\mathbf{x}^*) - f_M(\mathbf{x}_t)) \mathbb{1}[\ell_t = M] \leq \tilde{\mathcal{O}}\left(\sqrt{\gamma_{T_M(t)} T_M(t)} \log(T_M(t)/\delta)\right)$$

By the relationship between simple regret and the target-fidelity cumulative regret bounded above, we have the desired result.

$$\begin{aligned}
\mathrm{SR}(t) &= f_M(\mathbf{x}^*) - \max_{\mathbf{x}\in\mathbf{X}} f_M(\mathbf{x}) \\
&\leq \frac{1}{T} \sum_{t=1}^{T} (f_M(\mathbf{x}^*) - f_M(\mathbf{x}_t)) \mathbb{1}[\ell_t = M] \\
&\leq \tilde{\mathcal{O}}\left(\sqrt{\frac{\gamma_{T_M(t)}}{T_M(t)}} \log(T_M(t)/\delta)\right)
\end{aligned}$$

□

A.2 Proof of Theorem 2

Theorem 2 reveals the relationship between the simple regret of the $\tilde{f}_M$ and the simple regret of f_M.

Proof: First, by the definition of the expected excess risk, when using MAE as the loss function, we have

$$\mathbb{E}\left[|f_M - \tilde{f}_M|\right] - \mathcal{L}(f_M, \tilde{f}_M^*) = \mathrm{Rate}_{MF}(M, T).$$

Given Assumption 4, we have that

$$\mathbb{E}\left[|f_M(\mathbf{x}) - \tilde{f}_M(\mathbf{x})|\right] = \mathrm{Rate}_{MF}(M, T)$$

When $t = 1$, since the algorithm conducts random sampling on $\mathbf{X}$, we have

$$\mathrm{SR}_{\mathrm{MA}}(t) = \mathbb{E}\left[f_M(\mathbf{x}) - f_M(\mathbf{x}^*)\right] - \mathbb{E}\left[f_M(\mathbf{x}) - \tilde{f}_M(\mathbf{x})\right] \leq \mathrm{SR}(t) + \mathrm{Rate}_{MF}(M, t)$$

When $t > 1$, by the construction of $\hat{\mathbf{X}}_t$, we could similarly prove that the global optimum for f_M is contained in $\hat{\mathbf{X}}_t$. Then by enlarging Theorem 1 with the additional term $\mathrm{Rate}_{MF}(M, T)$, we arrive at the desired results. □

A.3 Proof of Theorem 3

Proof: With aforementioned assumptions, for $\forall 1 < \ell \leq M$, the learning on fidelity $\ell - 1$ and ℓ meets the assumption of Theorem 10 of Robinson et al. (2020). Then by Theorem 10 of Robinson et al. (2020), we have that for two consecutive fidelities $\ell - 1$ and ℓ, the convergence rate of the expected excess risk is bounded by

$$\text{Rate}_{MF}(\ell, t) \leq O\left(\text{Rate}_{MF}(\ell-1, t) + \frac{\sqrt{\left(\log_{T_\ell(t)} \text{Rate}_{MF}(\ell-1, t) + 1\right) \log T_\ell(t)}}{T_\ell(t)}\right)$$

By recursively applying Theorem 10 for $\ell = 2 \dots M$, we obtain the desired results. □

B Experiment Setup

In this section, we provide additional details about the experiment setup.

B.1 Implementation

The implementation is available at https://github.com/SchroDeCat/rMFBO.

B.2 Datasets

B.2.1 Augmented Rastrigin Test Function

The Augmented Rastrigin test function is a 1-dimensional synthetic test function for multi-fidelity optimization, defined over the domain $[-10, 10] \times [0, 1]$, where the last dimension represents the fidelity parameter. The function is described as follows:

Definition The base function $B(x)$ is given by:

$$B(x) = -\left(x^2 - 10\cos(2\pi x) + 10\right)$$

Low fidelity approximations The low fidelity approximations of the function are perturbed by adding normally distributed noise:

- $y_1 = B(x) + \text{np.random.normal}(1 \cdot |x|)$
- $y_2 = B(x) + \text{np.random.normal}(2 \cdot x^2, 8)$
- $y_3 = B(x) + \text{np.random.normal}(-0.2 \cdot x^4, 8)$

Properties

- The minimal value of $B(x)$ ($B_{\min} \approx 40.2$) is found at the sides of the domain.
- The maximum value of $B(x)$ ($B_{\max} = 0$) is found in the middle of the domain.

Discrete fidelities The fidelity parameter can take discrete values from the set:

$$\text{Fidelities} = \{0.1, 0.5, 0.75, 1\}$$

B.2.2 Augmented Rastrigin 20D Test Function

Similarly, the Augmented Rastrigin 20D test function is a multi-dimensional (20-dimensional) synthetic test function for multi-fidelity optimization, defined over the domain $[-5, 5]^{20} \times [0, 1]$, where the last dimension represents the fidelity parameter. The function is described as follows:

Definition The base function $B(x)$ for a single dimension is given by:

$$B(x) = -\left(x^2 - 10\cos(2\pi x) + 10\right)$$

Low fidelity approximations The low fidelity approximations of the function are perturbed by adding normally distributed noise:

- $y_1 = B(x) + \text{np.random.normal}(1 \cdot |x|)$
- $y_2 = B(x) + \text{np.random.normal}(2 \cdot x^2, 8)$
- $y_3 = B(x) + \text{np.random.normal}(-0.2 \cdot x^4, 8)$

Properties

- The maximum value of $B(x)$ ($B_{\max} = 0$) is found in the middle of the domain.

Discrete fidelities The fidelity parameter can take discrete values from the set:

$$\text{Fidelities} = \{0.1, 0.5, 0.75, 1\}$$

B.2.3 YahooGYM 4D Numeric Benchmark: RPART

The YahooGYM 4D Numeric benchmark is a multi-dimensional optimization problem with 4 numeric input parameters, across 4 instances, and 12 objectives. The fidelity of the benchmark is defined by the fraction of training data.

Instances The benchmark includes 4 different instances, with the default instance being 1489. The full list of instances is as follows:

```
['40981', '41146', '1489', '1067']
```

Objectives The benchmark evaluates 12 objectives:

- Mean Misclassification Error (`mmce`)
- F1 Score (`f1`)
- Area Under Curve (`auc`)
- Logarithmic Loss (`logloss`)
- RAM Usage during Training (`ramtrain`)
- RAM Usage for Model Storage (`rammodel`)
- RAM Usage during Prediction (`rampredict`)
- Training Time (`timetrain`)
- Prediction Time (`timepredict`)
- Misclassification Error Cost (`mec`)
- Index of Agreement (`ias`)
- Number of Features (`nf`)

For optimization, the objective `auc` is selected to be maximized.

Discrete train sizes The discretized fractions of training data range from 0 to 1, with a default value of 0.525.

Input parameters The input parameters for the benchmark are:

- `task_id`: Type: Constant, Value: 1489
- `cp`: Type: UniformFloat, Range: $[0.0001, 1.0]$, Default: 0.01, on log-scale
- `maxdepth`: Type: UniformInteger, Range: $[1, 30]$, Default: 16
- `minbucket`: Type: UniformInteger, Range: $[1, 100]$, Default: 50
- `minsplit`: Type: UniformInteger, Range: $[1, 100]$, Default: 50
- `trainsize`: Type: UniformFloat, Range: $[0.03, 1.0]$, Default: 0.525

B.2.4 YahooGYM 7D Numeric Benchmark: LCBench

The YahooGYM 7D Numeric benchmark is a multi-dimensional optimization problem with 7 numeric input parameters, across 34 instances, and 6 objectives. The fidelity of the benchmark is defined by the number of epochs.

Instances The benchmark includes 34 different instances, with the default instance being 3945. The full list of instances is as follows:

```
['3945', '7593', '34539', '126025', '126026', '126029', '146212',
'167104', '167149', '167152', '167161', '167168', '167181',
'167184', '167185', '167190', '167200', '167201', '168329',
'168330', '168331', '168335', '168868', '168908', '168910',
'189354', '189862', '189865', '189866', '189873', '189905',
'189906', '189908', '189909']
```

Objectives The benchmark evaluates 6 objectives:

- Time (`time`)
- Validation Accuracy (`val_accuracy`)
- Validation Cross Entropy (`val_cross_entropy`)
- Validation Balanced Accuracy (`val_balanced_accuracy`)
- Test Cross Entropy (`test_cross_entropy`)
- Test Balanced Accuracy (`test_balanced_accuracy`)

For optimization, the objective `test_balanced_accuracy` is selected to be maximized.

Discrete fidelities The discretized fidelities range from 1 to 52, with a default value of 26. The specific fidelities used are:

$$\{1, 8, 16, 32\}$$

Input parameters The input parameters for the benchmark are:

- `batch_size`: Type: UniformInteger, Range: $[16, 512]$, Default: 91, on log-scale
- `learning_rate`: Type: UniformFloat, Range: $[0.0001, 0.1]$, Default: 0.003162278, on log-scale
- `max_dropout`: Type: UniformFloat, Range: $[0.0, 1.0]$, Default: 0.5

- `max_units`: Type: UniformInteger, Range: $[64, 1024]$, Default: 256, on log-scale
- `momentum`: Type: UniformFloat, Range: $[0.1, 0.99]$, Default: 0.545
- `num_layers`: Type: UniformInteger, Range: $[1, 5]$, Default: 3
- `weight_decay`: Type: UniformFloat, Range: $[0.00001, 0.1]$, Default: 0.050005
- `epoch`: Type: UniformInteger, Range: $[1, 52]$, Default: 26

Reference Further details and information can be found in the referred publication:

https://www.tnt.uni-hannover.de/papers/data/1459/2020_arXiV_Auto_PyTorch.pdf

B.2.5 YahooGYM 13D Numeric Benchmark: XGBoost

The YahooGYM 13D Numeric benchmark is a multi-dimensional optimization problem with 13 numeric input parameters, across 4 instances, and 12 objectives. The fidelity of the benchmark is defined by the fraction of training data.

Instances The benchmark includes 4 different instances, with the default instance being 1489. The full list of instances is as follows:

```
['40981', '41146', '1489', '1067']
```

Objectives The benchmark evaluates 12 objectives:

- Mean Misclassification Error (`mmce`)
- F1 Score (`f1`)
- Area Under Curve (`auc`)
- Logarithmic Loss (`logloss`)
- RAM Usage during Training (`ramtrain`)
- RAM Usage for Model Storage (`rammodel`)
- RAM Usage during Prediction (`rampredict`)
- Training Time (`timetrain`)
- Prediction Time (`timepredict`)
- Misclassification Error Cost (`mec`)
- Index of Agreement (`ias`)
- Number of Features (`nf`)

For optimization, the objective `auc` is selected to be maximized.

Citation The benchmark data can be cited similarly as in the YahooGYM dataset.

Discrete train sizes The discretized fractions of training data range from 0 to 1, with a default value of 0.525.

Input parameters The input parameters for the benchmark are:

- `task_id`: Type: Constant, Value: 1489
- `booster`: Type: Categorical, Choices: {`gblinear`, `gbtree`, `dart`}, Default: `dart` (to allow all hyperparameters to be active)
- `alpha`: Type: UniformFloat, Range: $[0.0001, 999.9999]$, Default: 0.316227766, on log-scale
- `colsample_bylevel`: Type: UniformFloat, Range: $[0.01, 1.0]$, Default: 0.505
- `colsample_bytree`: Type: UniformFloat, Range: $[0.01, 1.0]$, Default: 0.505
- `eta`: Type: UniformFloat, Range: $[0.0001, 1.0]$, Default: 0.01, on log-scale
- `gamma`: Type: UniformFloat, Range: $[0.0001, 6.9999]$, Default: 0.0264575131, on log-scale
- `lambda`: Type: UniformFloat, Range: $[0.0001, 999.9999]$, Default: 0.316227766, on log-scale
- `max_depth`: Type: UniformInteger, Range: $[1, 15]$, Default: 8
- `min_child_weight`: Type: UniformFloat, Range: $[2.718281828459045, 149.9999]$, Default: 20.1926292064, on log-scale
- `nrounds`: Type: UniformInteger, Range: $[3, 2000]$, Default: 77, on log-scale
- `rate_drop`: Type: UniformFloat, Range: $[0.0, 1.0]$, Default: 0.5
- `skip_drop`: Type: UniformFloat, Range: $[0.0, 1.0]$, Default: 0.5
- `subsample`: Type: UniformFloat, Range: $[0.1, 1.0]$, Default: 0.55
- `trainsize`: Type: UniformFloat, Range: $[0.03, 1.0]$, Default: 0.525

Parameter conditions The conditions for the input parameters are defined as follows:

```
colsample_bylevel | booster in {'dart', 'gbtree'}
colsample_bytree | booster in {'dart', 'gbtree'}
eta | booster in {'dart', 'gbtree'}
gamma | booster in {'dart', 'gbtree'}
max_depth | booster in {'dart', 'gbtree'}
min_child_weight | booster in {'dart', 'gbtree'}
rate_drop | booster == 'dart'
skip_drop | booster == 'dart'
```

We fix the booster to `dart` to allow all hyperparameters to be active.

B.2.6 Antibody Binding Free Energy Benchmark

The Antibody Binding Free Energy Benchmark estimates the change in binding free energy ($\Delta\Delta G$) at the interface between each of 71,769 modified antibodies and the SARS-CoV-2 spike protein, as compared to the single reference antibody from which they are derived. These calculations were executed using supercomputing resources and are part of an antibody design process (Desautels et al., 2020, 2022).

Binding Free Energy Calculation The binding free energy ($\Delta\Delta G$) estimations were conducted using protein-structure-based simulation software (Das and Baker, 2008; Barlow et al., 2018; Schymkowitz et al., 2005; Sapozhnikov et al., 2023; Buß et al., 2018; DeBartolo et al., 2014). The low-fidelity calculations presented use STATIUM (DeBartolo et al., 2014), which compares local geometry around a mutation with a broad database of protein protein interfaces and operates on the principle that more favorable interactions should be more frequent to derive a score. The mid- and higher-fidelity calculations use the FoldX (Schymkowitz et al., 2005) and Rosetta Flex (Barlow et al., 2018) protocols, which modify the positions of protein side chains, calculate an average interface binding free energy ΔG, and compare with the unmutated version to calculate the mutational change in binding free energy, $\Delta\Delta G$. Lower $\Delta\Delta G$ values indicate stronger binding affinity between protein structures. Each calculation required several CPU hours, reflecting the computational intensity of the process. Here we take the negative values so that we need to maximize the objective during optimization.

Features The input sequence and structure data for these calculations is represented by an 86-dimensional feature vector, describing changes in the antibody-target interface relative to a reference sequence. These features capture structural and biochemical variations within the context of antibody design campaigns for derivatives of the s230 antibody targeting the SARS-CoV-2 spike protein.

Fidelities Binding free energy ($\Delta\Delta G$) calculations were performed at three fidelity levels, reflecting varying levels of computational resource allocation and prediction accuracy: **STATIUM**, **FoldX**, and **Rosetta**. Each tool computes $\Delta\Delta G$ as the mutational change (Δ) in binding free energy (ΔG) based on a single mutant sequence and a co-complex structure of the antibody-antigen pair.

These fidelity levels—**0** (STATIUM), **1** (FoldX), and **2** (Rosetta)—enable a hierarchical approach to binding free energy estimation. STATIUM delivers rapid, coarse-grained results; FoldX balances speed and precision; and Rosetta provides the most accurate, detailed insights into binding dynamics. The costs are specified as 0.2, 1.2, and 2.2 correspondingly.

B.3 Hyperparameter Choices

The following summarizes the key hyperparameter choices for the algorithms in our experiments.

- We set $\beta^{1/2} = 3$ for experiments except $\beta^{1/2} = 0.05$ for protein design task;
- For random discretization of MF-MES, we set the sample number to be 40000 and applied the Sobel engine for sampling.
- For the Monte Carlo approximation in (r)MF-MES and (r)MF-KG, we set the sample number to be 512;
- For the robust baselines proposed by Mikkola et al. (2023), we follow the previous practice by setting the threshold $c_1 = c_2 = 0.5$;
- For the kernel choices, we applied the kernel designed specifically for multi-fidelity hyperparameter tuning (Wu et al., 2020a) for (r)MF-MES and (r)MF-KG. For the deep kernel, we applied the RBF kernel (Srinivas et al., 2009; Rasmussen and Williams, 2006) as the base kernel and applied the backpropagation of Deep kernel learning (Wilson et al., 2016) to optimize the neural network and kernel parameters;
- For the iterative spectral norm estimation (Van Amersfoort et al., 2021), we set the iteration to be 100 for sufficient accuracy.
- In the deep kernel, the neural network named `LargeFeatureExtractor` consists of the following layers: a fully connected linear layer (`linear1`) with input dimension `data_dim` and output dimension 128 using data type `torch.float64` (`Linear(in_features=data_dim, out_features=128, dtype=torch.float64)`), followed by a ReLU activation function (`ReLU()`), and another fully connected linear layer (`linear4`) with input dimension 128 and output dimension $\max(\texttt{data\_dim} - 1, 2)$ using data type `torch.float64` (`Linear(in_features=128, out_features=max(data_dim-1, 2), dtype=torch.float64)`). This network extracts features from input data in the sequence: `linear1`, `relu1`, and `linear4`.

C Additional Results

We present additional results in this section. The $S1$, $S2$, and End denotes three time points during the optimization process where $S1 < S2 < End$. The results are collected from at least ten independent trials.

	Method	Dataset					
		Rastrigin-1D	Rastrigin-20D	RPART-4D	LCBench-7D	XGBoost-13D	Protein-86D
S1	RMFBO-DP	**5.79** ± 2.24	**103.12** ± 7.62	**0.28** ± 0.07	0.28 ± 0.01	0.18 ± 0.04	**4.57** ± 0.91
	MF-MES	30.06 ± 9.04	110.00 ± 0.00	0.38 ± 0.08	0.28 ± 0.01	0.16 ± 0.01	6.61 ± 0.49
	MF-KG	9.90 ± 4.11	110.00 ± 0.00	0.34 ± 0.04	0.32 ± 0.07	0.67 ± 0.12	5.64 ± 0.89
	SF-DK	33.58 ± 11.06	104.29 ± 5.35	0.29 ± 0.05	0.27 ± 0.02	0.17 ± 0.02	6.30 ± 0.75
	SF-MES	29.36 ± 8.94	109.86 ± 0.25	0.29 ± 0.06	**0.26** ± 0.01	**0.15** ± 0.01	6.50 ± 0.60
	SF-KG	7.48 ± 4.17	109.49 ± 0.94	0.32 ± 0.08	**0.26** ± 0.01	**0.15** ± 0.01	5.96 ± 0.71
	rMF-MES	20.16 ± 10.50	109.93 ± 0.13	0.39 ± 0.07	0.29 ± 0.02	0.18 ± 0.07	4.76 ± 1.01
	rMF-KG	6.30 ± 3.16	109.85 ± 0.27	0.38 ± 0.08	**0.26** ± 0.01	0.69 ± 0.34	5.51 ± 1.71
	Random	15.08 ± 6.35	110.00 ± 0.00	0.32 ± 0.07	0.30 ± 0.02	**0.15** ± 0.01	15.75 ± 0.70
	MF-UCB	12.50 ± 5.11	107.50 ± 3.43	0.31 ± 0.05	0.29 ± 0.02	0.16 ± 0.02	10.87 ± 1.02
	DNN-MFBO	28.50 ± 8.50	110.00 ± 0.00	0.36 ± 0.07	0.27 ± 0.01	0.16 ± 0.01	6.55 ± 0.59
S2	RMFBO-DP	**2.13** ± 1.47	**97.08** ± 8.29	0.25 ± 0.05	0.26 ± 0.01	**0.14** ± 0.01	**3.09** ± 0.98
	MF-MES	24.48 ± 8.38	108.16 ± 3.43	0.33 ± 0.09	0.28 ± 0.01	0.15 ± 0.01	4.99 ± 0.66
	MF-KG	3.51 ± 1.36	106.03 ± 6.96	0.27 ± 0.03	0.27 ± 0.01	0.67 ± 0.12	4.91 ± 0.98
	SF-DK	33.07 ± 11.42	97.63 ± 5.95	**0.24** ± 0.04	0.26 ± 0.01	**0.14** ± 0.02	5.50 ± 1.23
	SF-MES	27.09 ± 9.77	108.68 ± 1.98	0.25 ± 0.01	0.26 ± 0.01	**0.14** ± 0.01	5.50 ± 0.73
	SF-KG	3.44 ± 2.74	109.49 ± 0.94	**0.22** ± 0.03	0.26 ± 0.01	**0.14** ± 0.01	4.02 ± 1.18
	rMF-MES	15.42 ± 11.18	101.40 ± 9.02	0.27 ± 0.04	0.26 ± 0.01	**0.14** ± 0.02	4.76 ± 1.01
	rMF-KG	3.86 ± 1.23	107.18 ± 4.95	0.23 ± 0.02	**0.25** ± 0.01	0.69 ± 0.34	4.41 ± 1.50
	Random	7.07 ± 2.61	108.60 ± 1.76	0.23 ± 0.02	0.28 ± 0.01	**0.14** ± 0.01	14.77 ± 0.43
	MF-UCB	5.63 ± 2.11	107.50 ± 3.43	0.24 ± 0.03	0.27 ± 0.01	0.15 ± 0.02	9.98 ± 1.50
	DNN-MFBO	22.50 ± 7.04	107.91 ± 6.69	0.31 ± 0.08	0.27 ± 0.01	0.15 ± 0.01	4.83 ± 0.59
End	RMFBO-DP	**0.75** ± 0.30	**86.64** ± 9.80	**0.21** ± 0.03	**0.25** ± 0.01	**0.12** ± 0.01	**1.63** ± 0.81
	MF-MES	15.37 ± 6.83	105.66 ± 4.25	0.31 ± 0.08	0.28 ± 0.01	0.14 ± 0.01	4.09 ± 0.73
	MF-KG	1.84 ± 0.61	106.03 ± 6.96	0.24 ± 0.01	0.27 ± 0.01	0.67 ± 0.12	4.34 ± 0.85
	SF-DK	32.35 ± 11.98	87.28 ± 7.16	**0.21** ± 0.02	**0.25** ± 0.01	**0.12** ± 0.02	5.29 ± 1.08
	SF-MES	23.74 ± 9.98	100.92 ± 6.17	0.25 ± 0.01	**0.25** ± 0.01	0.14 ± 0.01	4.25 ± 1.56
	SF-KG	1.73 ± 0.84	108.20 ± 1.72	**0.22** ± 0.03	**0.25** ± 0.01	0.13 ± 0.01	3.25 ± 1.04
	rMF-MES	8.33 ± 7.36	100.93 ± 8.87	0.22 ± 0.03	0.26 ± 0.01	0.13 ± 0.02	4.29 ± 0.78
	rMF-KG	3.02 ± 1.18	103.37 ± 6.41	0.23 ± 0.02	**0.25** ± 0.01	0.69 ± 0.34	4.04 ± 1.18
	Random	5.05 ± 2.43	104.06 ± 6.15	0.22 ± 0.02	0.27 ± 0.01	0.13 ± 0.01	14.42 ± 0.33
	MF-UCB	5.63 ± 2.11	107.50 ± 3.43	0.22 ± 0.02	0.26 ± 0.01	0.13 ± 0.02	8.44 ± 1.20
	DNN-MFBO	20.50 ± 6.01	107.91 ± 6.69	0.29 ± 0.07	0.27 ± 0.01	0.14 ± 0.01	4.41 ± 0.52

Table 2: The simple regret of different methods on different datasets, at three selected costs. The best performance is emphasized in bold.

This updated table includes the results for DNN-MFBO, with values similar to those of MF-MES across all datasets and sections. As shown in the table, RMFBO-DP outperforms the baselines in most cases and at least match the baselines' performance, especially when in the later stage of the optimization. The results demonstrate the effectiveness of RMFBO-DP in multi-fidelity optimization.