Benchmarking the Performance of Bayesian Optimization across Multiple Experimental Materials Science Domains

by

Qiaohao Liang

B.S. Materials Science and Engineering
University of California, Berkeley, 2019
Submitted to the Department of Materials Science and Engineering in partial fulfillment of the requirements for the degree of Master of Science in Materials Science and Engineering at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY September 2021
© Massachusetts Institute of Technology 2021. All rights reserved.

Author ........................................
Department of Materials Science and Engineering
September 1, 2021

Certified by ........................................
Tonio Buonassisi
Professor, MIT Department of Mechanical Engineering
Thesis Supervisor

Certified by ........................................
Dr. John Fisher III
Principal Investigator, MIT CSAIL
Thesis Supervisor

Certified by ........................................
Rafael Gomez-Bombarelli
Professor, MIT Department of Materials Science and Engineering
Thesis Reader

Accepted by ........................................
Frances Ross
Professor, MIT Department of Materials Science and Engineering
Departmental Committee on Graduate Studies
Benchmarking the Performance of Bayesian Optimization across Multiple Experimental Materials Science Domains

by

Qiaohao Liang

Submitted to the Department of Materials Science and Engineering on September 1, 2021, in partial fulfillment of the requirements for the degree of Master of Science in Materials Science and Engineering

Abstract

Traditionally, experimental materials optimization has used design of experiments or intuition, combined with in-depth characterization. While these methods have obtained success over the years, they are facing increasing challenges today in the face of complex aggregated systems with larger design spaces. The materials objectives for these systems, e.g. environmental stability of solar cells or toughness of 3D printed mechanical structures, are typically costly to simulate and slow to experimentally evaluate. The need to shorten lab-to-market time of functional materials has inspired the use of machine learning and automation in materials optimization. Active learning algorithms, such as Bayesian Optimization (BO), have been leveraged for guiding autonomous high-throughput experimentation (HTE) systems. There have been individual studies successfully applying BO in experimental materials optimization, yet very few evaluated the performance of BO as a general optimization algorithm across a broad range of materials science domains.

In this work, we benchmark the performance of BO algorithms with a collection of surrogate model and acquisition function pairs across five diverse experimental materials systems, including carbon nanotube polymer blends, silver nanoparticles, lead-halide perovskites, as well as additively manufactured polymer structures and shapes. By defining acceleration and enhancement performance metrics as general materials optimization objectives, we find that for surrogate model selection, Gaussian Process (GP) with anisotropic kernels (automatic relevance detection, ARD) and Random Forests (RF) have comparable performance and both outperform the commonly used GP without ARD. We discuss the implicit distributional assumptions of RF and GP, and the benefits of using GP with anisotropic kernels in detail. We provide practical insights for experimentalists on surrogate model selection of BO during materials optimization campaigns.

Thesis Supervisor: Tonio Buonassisi
Title: Professor, MIT Department of Mechanical Engineering
Thesis Supervisor: Dr. John Fisher III
Title: Principal Investigator, MIT CSAIL
Acknowledgments

I would like to start by thanking my academic advisors Prof. Tonio Buonassisi and Dr. John Fisher for their outstanding mentoring and guidance. Tonio’s vision in accelerating materials research has always inspired me to work at the intersection of machine learning and materials science research. His all-around support as an advisor has been significant in the past two years for me at MIT. He has also shared his remarkable collaboration network across US and Singapore, creating great opportunities for my research work even when labs were affected by the pandemic. Besides his unmatched humor, John has been offering great suggestions on approaching machine learning problems, clarifying concepts, and constantly encouraging me to take on more difficult challenges in the field. I am also fortunate to have Prof. Rafael Goméz Bombarelli as my thesis reader from DMSE.

I would like to thank to all my colleagues in the MIT Photovoltaics Laboratory and the Sensing Learning and Inference Laboratory at MIT. Specifically, I am grateful to Dr. Armi Tiihonen and Dr. Liu Zhe for the advice and senior support in the past two years. Interactions with Shiqi, Titan, Alex, Richa, Felipe, Erin, Nithin have also been very inspiring. I would like to also thank Dr. Zekun Ren (Singapore and MIT Alliance of Research and Technology), Dr. Aldair E. Gongora (Boston University), and collaborators from US Air Force Laboratories, A*STAR, and National University of Singapore, who have been crucial in supporting the benchmarking study presented below.

Last but not least, I am immensely grateful to my parents. Your academic journeys and dedication in life have always been something I looked up to. In the ups and downs of this marvelous journey, thank you for being with me and sharing your love. The past two years at MIT have been full of adversities but also very rewarding. I have found resilience, faith and dedication within myself, and am ready for the next challenge ahead.
Contents

1 Introduction 6

2 Data and Methods 9
   2.1 Experimental materials datasets 9
   2.2 Bayesian Optimization 10
   2.3 Prediction by surrogate models 13
   2.4 Pool-based active learning framework 14
   2.5 Benchmarking metrics and baselines 17

3 Results 20
   3.1 Observation of performance through case study on Crossed barrel dataset 20
   3.2 Comparison of performance across datasets 24

4 Discussion 28
   4.1 Comparison of RF and GP as surrogate models 28
   4.2 Benefits of using GP with anisotropic kernels 32

5 Conclusions 35

6 Appendix 37
Chapter 1

Introduction

Autonomous experimental systems have recently emerged as the new frontier for accelerated materials research. These systems excel at optimizing materials objectives, e.g. environmental stability of solar cells or toughness of 3D printed mechanical structures, that are typically costly, slow, or difficult to simulate and experimentally evaluate. While autonomous experimental systems are often associated with high sample synthesis rates via high-throughput experiments (HTE), they may also utilize closed-loop feedback from machine learning (ML) during materials property optimization. The latter has motivated integration of advanced lab automation components with ML algorithms. Specifically, active learning algorithms have traditionally been applied to minimizing total experiment costs while maximizing machine learning model accuracy through hyperparameter tuning. Their primary utility for materials science research, where experiments remain relatively costly, lies in an iterative formulation that proposes targeted experiments with regard to a specific design objective based on prior experimental observations. Bayesian optimization (BO), one class of active-learning learning methods, utilizes surrogate model to approximate a mapping from experiment parameters to an objective criterion, and provides optimal experiment selection when combined with an acquisition function. BO has been shown to be a data-efficient closed-loop active learning method for navigating complex design spaces. Consequently, it has become an appealing methodology for accelerated materials research and optimizing material
properties [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22] beyond state-of-the-art.

The materials science community has seen successful demonstrations in performing materials optimization via autonomous experiments guided by BO and its variants [17, 23, 24, 25, 26]. Naturally, previous work emphasized the ability to achieve materials optimization with fewer experimental iterations. There have been very few quantitative analyses of the acceleration or enhancement resulting from applying BO algorithms and discussions on sensitivity of BO performance to surrogate model and acquisition function selection. Rohr et al. [28], Graff et al. [29] and Gongora et al. [24] have evaluated the performance of BO using multiple surrogate models and acquisition functions within specific electrocatalyst, ligand and mechanical structure design spaces respectively. However, comprehensive benchmarking of the performance of BO algorithms across a broad array of experimental materials systems, as presented in this work, has not been done. Although one could test BO across various analytical functions or emulated materials design spaces [30, 25], empirical performance evaluation on a broader collection of experimental materials science data is still necessary to provide practical guidelines. Optimization algorithms need systematic and comprehensive benchmarks to evaluate their performance, and lack of these could significantly slow down advanced algorithm development, eventually posing obstacles for building full autonomous platforms. Presented below, the benchmarking framework, practical performance metrics, datasets collected from realistic noisy experiments, and insights derived from side-by-side comparison of BO algorithms will allow researchers to evaluate and select their optimization algorithm before deploying it on autonomous research platforms. Our work provides comprehensive benchmarks for optimization algorithms specifically developed for autonomous and high-throughput experimental materials research. Ideally, it provides insight for designing and deploying Bayesian optimization algorithms that suit the sample generation rate of future autonomous platforms and tackle materials optimization in more complex design spaces.

In this work, performance of BO is benchmarked across five different experimental materials science datasets, optimizing properties of carbon nanotube polymer blends, silver nanoparticles, lead-halide perovskites, and additively manufactured polymer
structures and shapes. We utilize a pool-based active learning framework to approximate experimental materials optimization processes. We also adapt metrics such as enhancement factor and acceleration factor to quantitatively compare performances of BO algorithms against that of a random sampling baseline. We observe that when paired with the same acquisition functions, Random Forest (RF) \cite{31,32,33} as a surrogate model can compete with Gaussian Process (GP) \cite{4} with automatic relevance detection (ARD) \cite{34} that has anisotropic kernels. They also both outperform commonly used GP without ARD that has isotropic kernels. Our discussion on the differences in the implicit distributional assumptions of surrogate models and the benefits of using GP with anisotropic kernels yield deeper insights regarding surrogate model selection for materials optimization campaigns. These discussions suggest guidelines on using BO for general materials optimization. We also offer open source implementation of benchmarking code and datasets to support future development of such algorithms in the field.
Chapter 2

Data and Methods

2.1 Experimental materials datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Synthesis</th>
<th>Size</th>
<th>( n_{\text{dim}} )</th>
<th>Optimization Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3HT/CNT [35]</td>
<td>Composite blends</td>
<td>Drop casting</td>
<td>178</td>
<td>5</td>
<td>Electrical conductivity</td>
</tr>
<tr>
<td>AgNP [36]</td>
<td>Silver nanoparticles</td>
<td>Flow synthesis</td>
<td>164</td>
<td>5</td>
<td>Abs. spectrum score</td>
</tr>
<tr>
<td>Perovskite [23]</td>
<td>Thin film perovskite</td>
<td>Spin coating</td>
<td>94</td>
<td>3</td>
<td>Stability score</td>
</tr>
<tr>
<td>Crossed barrel [24]</td>
<td>3D printed structure</td>
<td>3D printing</td>
<td>600</td>
<td>4</td>
<td>Mechanical toughness</td>
</tr>
<tr>
<td>AutoAM [37]</td>
<td>Materials manufacturing</td>
<td>3D printing</td>
<td>100</td>
<td>4</td>
<td>Shape score</td>
</tr>
</tbody>
</table>

As seen in Table 1, I have assembled a list of five materials datasets with varying sizes, dimensions \( n_{\text{dim}} \), and materials systems. These diverse datasets are generated from autonomous experimental studies of collaborators, and facilitate BO performance analysis across a broad range of materials. They contain three to five independent input features, one property as materials optimization objective, and contain from a few tens to hundreds of data points. Based on their optimization objectives, the design space input features in the datasets range from materials compositions to synthesis processing parameters, as seen in Table T1 - T5 in Appendix. For consistency, each dataset has its optimization problem formulated as global minimization.

It should be noted that while all datasets were gathered from relatively high-throughput experimental systems, P3HT/CNT, AgNP, Perovskite, and AutoAM had
Figure 1: Experimental materials dataset design space manifold complexity visualization. (a) Histogram of objective values normalized to zero-mean without loss of generality. (b) Input feature space, i.e. design space, visualization after dimension reduction to 3D via principal component analysis (PCA). The colors of each point in the datasets indicate its value. PCA was performed to reduce all datasets dimensions to three for visualization, and the three axes shown are the top three principal component directions of each dataset.

BO guiding the selection of subsequent experiments partially through the materials optimization campaigns. Across the datasets, the differences in distribution of normalized objective values can be observed in Figure 1(a); the differences in distribution of sampled data points in its respective materials design space can be seen in Figure 1(b). The five materials datasets in the current study are available in the following GitHub repository [38].

2.2 Bayesian Optimization

Bayesian optimization [3 4 5] aims to solve the problem of finding a global optimum (min or max) of an unknown objective function $g$: $\vec{x}^* = \arg \min_{\vec{x}} g(\vec{x})$ where $\vec{x} \in X$ and $X$ is a domain of interest in $\mathbb{R}^{n_{\text{dim}}}$. BO holds the assumption that this black-box function $g$ can be evaluated at any $\vec{x} \in X$ and the responses are noisy point-wise observations $(\vec{x}, y)$, where $\mathbb{E}[y|g(\vec{x})] = g(\vec{x})$. The surrogate model $f$ is probabilistic and consists of a prior distribution that approximates the unknown objective function $g$, and is sequentially updated with collected data to yield a Bayesian posterior belief.
of \( g \). Decision policies aimed to reach the optimum with fewer experiments are implemented in acquisition functions, which use mean and variance predicted at any \( \tilde{x} \in X \) in the posterior to select the next observation to be performed.

The BO algorithm is comprised of both a surrogate model and an acquisition function. The surrogate models considered in this study are random forest (RF) [31], Gaussian process (GP) regression [39], and GP with automatic relevance detection (ARD) [39, 5, 34].

1. To approximate the experience of a researcher with little prior knowledge of a materials design space, for RF, I have chosen hyperparameters applicable across all five datasets without loss of generality: \( n_{\text{tree}} = 100 \) and bootstrap = True. Figure S1 in Appendix shows that \( n_{\text{tree}} = 100 \) is a suitable hyperparameter for RF surrogate models when applied to the five datasets.

2. For hyperparameters of GP, kernels are chosen from Matérn52, Matérn32, Matérn12, radial basis function (RBF), and multilayer perceptron (MLP). The initial lengthscale for each kernel was set to unit length.

3. For hyperparameters of GP ARD, the above kernel choices from GP kept, ARD is also used, which allows GP to keep anisotropic kernels. The kernel function of GP then has individual characteristic lengthscales \( l_j \) for each of the input feature dimensions \( j \).

As an example, in dimension \( j \), Matérn52 kernel function between two points \( \mathbf{p}, \mathbf{q} \) in design space would be

\[
k(p_j, q_j) = \sigma_0^2 \cdot \left(1 + \frac{\sqrt{5}r}{l_j} + \frac{5r^2}{3l_j^2}\right) \exp\left(-\frac{\sqrt{5}r}{l_j}\right) \tag{2.1}
\]

where \( r = \sqrt{(p_j - q_j)^2} \), \( \sigma \) is standard deviation and \( l_j \) is the characteristic lengthscale specifically for dimension \( j \). These characteristic lengthscales can be used to estimate the distance moved along \( j^{th} \) dimension from the input values in design space before the change of objective values become uncorrelated with this feature. \( \frac{1}{l_j} \) is thus useful in understanding the sensitivity of objective value to input feature \( j \).
The selected surrogate model is then paired with one of the acquisition functions. We tested three acquisition functions in our study, including expected improvement (EI), probability of improvement (PI), and lower confidence bound (LCB).

\[
\text{EI}(\mathbf{x}) = (y_{\text{best}} - \hat{\mu}(\mathbf{x}) - \xi) \cdot \Phi(Z) + \hat{\sigma}(\mathbf{x}) \varphi(Z)
\]

(2.2)

\[
\text{PI}(\mathbf{x}) = \Phi(Z)
\]

(2.3)

where

\[
Z = \frac{y_{\text{best}} - \hat{\mu}(\mathbf{x}) - \xi}{\hat{\sigma}(\mathbf{x})}
\]

(2.4)

\(\hat{\mu}\) and \(\hat{\sigma}\) are the mean and standard deviation estimated by surrogate model; \(y_{\text{best}}\) is best discovered objective value within all collected values so far; \(\xi = 0.01\) is jitter value that can slightly control exploration and exploitation; \(\Phi\) and \(\varphi\) are the cumulative density function and probability density function of a normal distribution.

\[
\text{LCB}_\lambda(\mathbf{x}) = -\hat{\mu}(\mathbf{x}) + \bar{\lambda}\hat{\sigma}(\mathbf{x})
\]

(2.5)

where \(\bar{\lambda}\) is an adjustable ratio between exploitation and exploration. \(\bar{\lambda}\) between 0.1 and 10 are tested in the study, representing acquisition strategies from greedy to explorative.

In addition, these surrogate models, their hyperparameters, and acquisition functions were chosen because they represent the majority of off-the-shelf options accessible, and are ones that have been widely applied to materials optimization campaigns in the field. The presented work provides a comprehensive test across the five datasets in order to reflect how each BO algorithm, resulting from the pairing above, performs across many different materials science design spaces. GP and RF were also selected as examples to specifically illustrate how the differences in implicit distributional assumptions of surrogate models could affect their predictions of the mean and standard
deviation when selecting subsequent experiments and performance in BO.

2.3 Prediction by surrogate models

In order to estimate the mean $\hat{\mu}(\vec{x}_*)$ and standard deviation $\hat{\sigma}(\vec{x}_*)$ of predicted objective value at a previously undiscovered observation $\vec{x}_*$ in design space:

For a Gaussian process (GP), it assumes a prior over the design space that is constructed from already collected observations $(\vec{x}_i, y_i)$, $i = 1, 2, ..., n$. This prior is the source of implicit distributional assumptions, and when an undiscovered new observation $(\vec{x}_*, y_*)$ is being considered during noisy setting ($\sigma = 0.01$), the joint distribution between the objective values of collected data $\vec{y} \in \mathcal{R}^n$ and $y_*$ is

$$
\begin{bmatrix}
\vec{y} \\
y_*
\end{bmatrix} \sim \mathcal{N}
\begin{pmatrix}
0, \\
K_n + \sigma^2 I
\end{pmatrix}
\begin{bmatrix}
K + \sigma^2 I & K_* \\
K_* & K_{**}
\end{bmatrix}^{-1}
\begin{bmatrix}
\vec{y} \\
y_*
\end{bmatrix}
$$

(2.6)

$K$ is the covariance matrix of the input features $X = \{\vec{x}_i | i = 1, 2, ..., n\}$; $K_*$ is the covariance between the collected data and new input feature $\vec{x}_*$; $K_{**}$ is the covariance between the new data. For each of the covariance matrices, $K_{pq} = k(\vec{x}_p, \vec{x}_q)$, where $k$ is the kernel function, whether isotropic or anisotropic, used in GP.

Then from the posterior, estimates are

$$
\hat{\mu}(\vec{x}) = y_* = K_*[K + \sigma^2 I]^{-1}\vec{y}
$$

(2.7)

and covariance matrix

$$
cov(y_*) = K_{**} - K_*[K + \sigma^2 I]^{-1}K_*^T
$$

(2.8)

The standard deviation value $\hat{\sigma}(\vec{x}_*)$ can be obtained from the diagonal elements of this covariance matrix.

For a random forest (RF), let $\hat{h}_k(\vec{x}_*)$ denote the prediction of objective value from the $k^{th}$ decision tree in the forest, $k = 1, 2, ..., n_{\text{tree}}$. 

13
then
\[ \hat{\mu}(\vec{x}^*) = \frac{1}{n_{\text{tree}}} \sum_{k=1}^{n_{\text{tree}}} \hat{h}_k(\vec{x}^*) \] (2.9)

and
\[ \hat{\sigma}(\vec{x}^*) = \sqrt{\frac{\sum_{k=1}^{n_{\text{tree}}} (\hat{h}_k(\vec{x}^*) - \hat{\mu}(\vec{x}^*))^2}{n_{\text{tree}}}} \] (2.10)

The median or other variations could also be used in future studies to aggregate the predictions for potential improvement in robustness [40].

### 2.4 Pool-based active learning framework

Within each respective experimental dataset, the set of data points form a discrete representation of ground truth in the materials design space. Figure 2 shows the pool-based active learning benchmarking framework used to simulate materials optimization campaigns guided by BO algorithms in each materials system.

As seen in Figure 2, to approximate early stage exploration during each optimization campaign, \( n = 2 \) initial experiments are drawn randomly with no replacement from original pool \( D = \{(\vec{x}_i, y_i)|i = 1, 2, \ldots, N\} \) and add to collection \( X = \{(\vec{x}_i, y_i)|i = 1, 2, \ldots, n\} \). During planning stage, surrogate model \( f \) is used to estimate the mean \( \hat{\mu}(\vec{f}) \) and standard deviation \( \hat{\sigma}(\vec{f}) \). We then evaluate the acquisition function values \( \alpha(\hat{\mu}(\vec{f}), \hat{\sigma}(\vec{f})) \) for each remaining experimental action \( \vec{f} \in D \) in parallel. At each cycle, action \( \vec{f}^* = \arg \max_x \alpha(\vec{f}) \) will be selected as next experiment. During inference stage, after selecting action \( \vec{f}^* \), the corresponding sample observation \( y^* \) is obtained, and \( (\vec{f}^*, y^*) \) is added to \( X \) and removed from set \( D \). The new observation \( (\vec{f}^*, y^*) \) is added to collection \( X \) and the surrogate model is retrained on \( X \) and updated. The sequential alternation between planning and inference is repeated until undiscovered data points run out.

The framework has the following properties:
Figure 2: Benchmarking framework including a simulation of BO performing closed-loop optimization with alternating inference and planning stages. $X$ is the iteratively collected sequence of experimental data $(\vec{x}, y)$ during optimization campaign. $D$ is the original pool or total undiscovered set of data from which next experiments are selected. $f$ is the surrogate model used to estimate mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$, which parameterize the acquisition function $\alpha$ to select next experiment $\vec{x}^*$ to be evaluated.

1. It has the traits of an active learning study as it contains a machine learning model that is iteratively refined through subsequent experimental observation selection based on information from previously explored data points. The framework is also adapted for BO, and emphasizes optimization of materials objectives over building an accurate regression model in design space.

2. It is derived from pool-based active learning. Besides the randomly selected initial experiments, the subsequent experimental observations are selected from the total pool of undiscovered data points $(\vec{x}, y) \in D$, whose input features $\vec{x}$ are all made available for evaluation by the acquisition functions. The ground truth in the materials design space was represented with discrete data points over a continuous emulation for the following reasons:
(a) In real research scenarios, materials design spaces are not completely continuous due to noise and limitation in resolution of equipment apparatus and experiment design.

(b) Because many materials datasets do not cover their design space uniformly with high density, the fitted ground truth model would have greater variance in regions that were loosely covered by the regression experimental dataset. As a result, even if overfitting is not considered, the continuous emulator could have varied accuracy across its design space compared to real experimental ground truth, greatly affecting optimization results.

(c) To emulate materials design spaces, selecting of models such as GP introduces smoothness assumptions into the design space, and thus during the benchmarking process could give great advantages to BO algorithms with GP surrogate models sharing similar gaussianity assumptions. In Figure S2 - S3 from Appendix, I show how such induced bias from different ground truth models affects evaluation of the performance of BO.

3. At each learning cycle of the framework, instead of selecting a larger batch, only one new experiment is obtained. In our retrospective study, keeping a batch size of 1 allows us to directly compare the performance of BO algorithms across different materials studies, which have varying dimensions and dataset sizes. In real experimental setups, the exact tradeoff between batch size and cost of experiment parallelization should be determined by researchers and their equipment apparatus limitations.

Each BO algorithm is evaluated for 50 ensembles with 50 independent random seeds governing the initialization of experiments. The aggregated performance of BO algorithms, derived from 50 averaged runs resulting from 10 random five-fold splits using the 50 original ensembles, is compared against a statistical random search baseline, and its performance is quantitatively evaluated via active learning metrics defined in the sections below.
2.5 Benchmarking metrics and baselines

In Figure 3 and 4 below, some metrics and statistical baselines are introduced when benchmarking the performance of BO algorithms in the context of a pool-based active learning framework.

$$\text{Top\%}(i) = \frac{\text{number of top candidates discovered}}{\text{number of total top candidates}} \in [0, 1] \quad (2.11)$$

Top\%(i) is the fraction of the candidates with top 5% materials objective values discovered by cycle \(i = 1, 2, 3, \ldots, N\). For example in Figure 3(a), Top\%(i) is the fraction of the crossed barrel structures with top 5% toughness discovered by cycle \(i\).

Top\% describes how quickly can a BO guided autonomous experimental system could identify multiple top candidates in a materials design space. Keeping multiple well-performing candidates allows one to not only observe regions in design space that frequently yield high-performing samples but also have backup options for further evaluation should the most optimal candidate fail in subsequent materials applications evaluations. There are research objectives related to finding any good materials candidate, yet in those cases, random selection could outperform optimization algorithms due to luck in a simple design space. Our objective of finding multiple or all of top tier candidates is more applicable to experimental materials optimization scenario and suitable for demonstrating the true efficacy and impact of BO.

To quantify the acceleration of discovery from BO, I adapt two other metrics similar to the ones from Rohr et al. [28].

Both compared to a statistical random baseline, Enhancement Factor (EF)

$$\text{EF}(i) = \frac{\text{Top\%}_{\text{BO}}(i)}{\text{Top\%}_{\text{random}}(i)} \quad (2.12)$$

shows how much improvement in a metric one would receive at cycle \(i\),
and Acceleration Factor (AF)

\[ \text{AF}(\text{Top}%) = a = \frac{i_{\text{BO}}}{i_{\text{random}}} \]  

is the ratio of cycle numbers showing how much faster one could reach a specific value Top% of BO compared to random. 

For the random baseline in Figure 3(a), assuming a total pool of \( N \) data points and the number of good materials candidates \( M = 0.05 N \), at cycle \( i = 1 \), expected probability of finding a good candidate is \( P(1) = 0.05 \) and expected value of Top\(\%\)(1) = \( \frac{1 \cdot P(1)}{M} \) = 0.0016.

Then at cycle \( i = 2, 3, ..., N \), there is

\[ P(i) = \frac{M - \sum_{n=1}^{i-1} P(n)}{N - i} \]  

and

\[ \text{Top}\%(i) = \frac{\sum_{n=1}^{i} P(n)}{M} \]

Jaccard similarity, \( J = \frac{|A \cap B|}{|A \cup B|} \), is the size of the intersection divided by the size of the union of two finite sample sets. In Figure 3(d), Jaccard similarity index is used in this study to further investigate how similar two BO algorithms optimization paths were in the design space when they start from the same initial experiments and showed comparable performance. Using the same 50-ensemble runs that generated Figure 3(a), Jaccard similarity value \( J(i) \) can be calculated in Figure 3(d) at each learning cycle \( i \), where \( A(i) \) and \( B(i) \) are the two sets of data points sequential collected at each learning cycle during an optimization path guided by BO algorithms A and B.

Between two optimization paths starting with the same two initial data points:

1. The statistically most overlap happens when two paths are identical, resulting in \( J(i) = 1, i = 1, 2, ..., N \);
2. The statistically least overlap happens when the two follow drastically different paths until they run out of data points undiscovered by both algorithms, resulting in

\[
J(i) = \begin{cases} 
1 & 1 \leq x \leq 2 \\
\frac{1}{i - 1} & 3 \leq i \leq \frac{N}{2} + 1 \\
\frac{2i - N}{N} & \frac{N}{2} + 2 \leq i \leq N
\end{cases}
\]  \quad (2.16)
Chapter 3

Results

3.1 Observation of performance through case study on Crossed barrel dataset

While the five datasets covered a breadth of materials domains, the relative performances of tested BO algorithms were observed to be quite consistent. The benchmarking results are thus showcased using the Crossed barrel dataset \[24\], which was collected by grid sampling the design space through a robotic experimental system while optimizing the toughness of 3D printed crossed barrel structures. For the full combinatorial study including all types of GP kernels and acquisition functions, please kindly refer to Figure S5 - S9 in Appendix besides Figure 3.

Figure 3(a) illustrates learning rates based on Top\% metric and the following are observed:

1. RF initially excels at lower learning cycles, while GP with ARD takes the lead after Top\% = 0.46. Under the same acquisition function, the performance of RF as a surrogate model is often on par, if not slightly worse, when compared to the performance of GP with ARD.

2. Both GP with ARD and RF outclass GP without ARD.

3. LCB\(_T\) typically outperforms other acquisition functions LCB\(_X\) that overly em-
Figure 3: Aggregated performance of BO algorithms on the Crossed barrel dataset measured by (a) Top% vs. learning cycle $i$ against random baseline, and how (b) Enhancement factor EF and (c) acceleration factor AF are derived from it. The algorithms with GP as surrogate model are labeled in red, and RF in blue; higher color saturation is correlated with better performance. Variation at each learning cycle is visualized by plotting the median as well as shaded regions representing the 5th to 95th percentile of the aggregated 50-run ensembles. The acquisition functions used are EI and LCB$_2$ (d) Jaccard similarity index calculated between the optimization campaign sequences of BO algorithms RF: LCB$_2$ and GP ARD: LCB$_2$. The median, 5th, and 95th percentile of the 50-run ensemble are shown respectively.
phasize exploration or exploitation as seen in Figure S5 - S9 in Appendix, and is therefore used as representative LCB acquisition function for all surrogate models. LCB$_2$ at times even outperformed EI, which is a very popular acquisition function in many previous materials optimization studies but has also been known to make excessive greedy decisions [41, 42, 43]. The performance of BO algorithms using probability of improvement (PI) as acquisition function has also been evaluated, but its performance was quite consistently worse than EI and therefore not the focus of discussion; this observation can be partially attributed to PI only focusing on how likely is an improvement occurs at next experiment, but not considering how much improvement could be made during evaluation.

When trying to further compare the BO algorithms with different surrogate models in this work, I would like to keep the acquisition function consistent. Because of the observation above, the same acquisition function LCB$_2$ was used for discussions below.

We would like to highlight the relative performances of BO algorithms that utilize surrogate models GP ARD (Matérn52 kernel), RF, and GP (Matérn52 kernel). To quantify the relative performance, Top% = 0.8 is set as a realistic goal to indicate 80% of the structures with top 5% toughness (Figure 3a) have been identified. For surrogate models paired with LCB$_2$, it can be observed that GP with ARD and RF reach that goal by evaluating approximately 75 and 85 candidates out of the total of 600, whereas GP without ARD needs about 170 samples out of 600. Top% rises initially as slowly as the random baseline because the surrogate models suffer from high variance in prediction, having only been trained with a small datasets; Top% ramps up very quickly as the model learns to become more accurate in identifying general regions of interest to explore; the rate of learning eventually slows down at high learning cycles because the local exploitation for the global optimum has exhausted most if not all top 5% toughness candidates, and the algorithms therefore switch to exploring sub-optimal regions. Therefore, it can be assumed that the most valuable regions to examine performance is before each curve reaches Top% = 0.8 and Top% = 0.8 can be used as a realistic optimization goal.
The aggregated performance of BO algorithms is further quantified via EF and AF curves in Figure 3(b), 3(c): starting off with small EFs or AFs before the surrogate model gains more accuracy; reaching absolute EF$_{\text{max}}$ and AF$_{\text{max}}$ of up to $8 - 9 \times$. Eventually, the learning algorithms show diminishing returns from an information gain perspective as the runs progress deeper into the optimization campaigns during pool-based active learning. It can be observed that for the two BO algorithms both with same acquisition function LCB$_2$ but different surrogate model GP ARD and RF, they reach EF$_{\text{max}}$ at different learning cycles and AF$_{\text{max}}$ at different Top%, both corresponding to the switch of best performing algorithm around Top% = 0.46. RF: LCB$_2$ clearly excels at lower learning cycles, yet GP ARD: LCB$_2$ takes the lead and would reach Top% = 0.8 with fewer experiments. Therefore, these results objectively show that optimal BO algorithm selection varies with assigned experiment budget and specific optimization task [28].

Since RF: LCB$_2$ and GP ARD: LCB$_2$ were identified to have comparable performance, I wanted to further investigate how similar their optimization paths were in the design space when starting from the same initial experiments. In Figure 3(d), I use Jaccard similarity index to quantify the similarity between the optimizations paths of these two BO algorithms. As baselines, I have also drawn what the Jaccard similarity value would look like between two optimization paths that begin with same initial experiments and statistically have least overlap or most overlap. When $i = 1$ or 2, the same initial experiments are given to the two BO algorithms, and $J = 1$. When $2 < i < 18$, it can be seen that the Jaccard similarity value drops as quickly as the statistically least overlapping paths, indicating that despite the fact that GP with ARD and RF were trained on the same initial experiments at the onset, they follow very different paths in the materials design space. This behavior indicates that, despite achieving comparable final performance, they exploit the underlying physics differently by virtue of the choice of experiments.

When $i \geq 18$, the general trend is that $J$ increases with $i$, indicating that the paths chosen by the two algorithms gradually start to have some overlap as they move
towards finding crossed barrels structures with high toughness. Recall both algorithms reached \( \text{Top\%} = 0.8 \) between 75 to 85 learning cycles in Figure 3(a), and between those learning cycles, it is observed that \( J \) is approximately between 0.27 - 0.33, still considerably far from \( J = 1 \). This observation shows that while both algorithms have comparable performance in the task of finding crossed barrel structures with good toughness, due to their different choice of surrogate models, their paths towards discovering optimum can differ considerably.

In addition, the Jaccard similarity value does not increase monotonically, and a significant drop can be seen in \( J \) such as one around \( i = 50 \), which coincides with the learning cycles where GP ARD : LCB\(_2\) overtook RF : LCB\(_2\) as best performing algorithm in Figure 3(a). Since the two algorithms used the same acquisition function, this observation shows that while in general the optimization paths of the two algorithms have more overlap overtime, occasional divergence paths still take place because the two algorithms have considerable difference in gathered data used to learn their surrogate models and how their surrogate models predict mean and standard deviation. GP ARD : LCB\(_2\) and RF : LCB\(_2\) started at the same two initial experiments and use the same acquisition function, and the only difference is the surrogate model used. Thus, the divergence and convergence in optimization paths can be again primarily attributed to GP ARD and RF exploiting underlying physics of crossed barrel structure differently. Figure 3(d) highlights the impact of different surrogate model selection beyond final performance, and to provide better guidelines to future research, inspires us to further investigate the role of surrogate models.

### 3.2 Comparison of performance across datasets

To further assess the performance of BO, optimization campaigns were conducted for the P3HT/CNT, AgNP, AM ARES, and Perovskite datasets. Across most, if not all the investigated datasets, it was quite observed consistently that the performance of BO algorithms using GP with ARD and RF as surrogate models were comparable, and both outperform those using GP without ARD in most datasets. To illustrate, in
Figure 4: Normalized $\text{EF}_{\text{max}}$ demonstrated by BO algorithms having GP without ARD, GP with ARD, and RF as surrogate models and all using LCB$_7$ as acquisition function. In each dataset, the BO algorithm with the largest $\text{EF}_{\text{max}}$ had its EF scaled to 1, and the other two BO algorithms showing lower $\text{EF}_{\text{max}}$ were correspondingly scaled, resulting in five sets of column plots. For each algorithm applied across datasets, the median of $\text{EF}_{\text{max}}$ is shown by the barplots, and its 5$^{\text{th}}$ and 95$^{\text{th}}$ percentile are shown by respective floating bars.
Figure 4, I show such relative performance using normalized $EF_{\text{max}}$ of BO algorithms same acquisition function $\text{LCB}_T$ but with different surrogate models across all five datasets. In addition to the observation on relative performance, it is also observed that BO algorithms with RF and GP ARD as surrogate model also have plenty overlap between their $5^{th}$ to $95^{th}$ percentile across five datasets, further indicating their similarity in performance. We observe that GP with anisotropic kernels (GP ARD) is shown to be a great surrogate model across most materials domains, with RF being a close second, and both proving to be robust models for future optimization campaigns.

Notably, $EF_{\text{max}}$ of the other four datasets datasets were in the $2\times$ to $5\times$ range as seen in Figure S4 from Appendix, which is noticeably lower than the $EF_{\text{max}}$ of the crossed barrel dataset in Figure 3(b). The difference in the absolute $EF_{\text{max}}$ can be attributed to the data collection methodology of the individual datasets. While the crossed barrel dataset was collected using a grid sampling approach, the other four studies were collected along the path of a BO guided materials optimization campaign. Therefore, these four datasets were smaller in size and possessed an intrinsic enhancement and acceleration within their datasets. As a result, it is reasonable that these datasets demonstrate lower EFs, AFs during benchmarking. Nevertheless, the fact that both EFs and AFs are still larger than 1 indicate that further acceleration and enhancement is still possible when given design spaces as parts of optimization paths. The results enhance BO’s role as an experiment selection tool. Noticeably, the Perovskite dataset had the most intrinsic acceleration because its next experimental choice was guided by BO infused with probabilistic constraints generated from DFT proxy calculations of the environmental stability\cite{23} of perovskites. As a result, the optimization sequence to be chosen in that study is already narrowed down to a more efficient path from initial experiments to final optimum, making the random baseline to appear arbitrarily much worse. Another interesting observation is how the performance of GP without ARD (isotropic kernels) as surrogate model catches up with GP ARD and RF when the design space has an already "easier" path towards the optimum. That is, when materials design space is relatively simple, GP without
ARD can serve as an equally good surrogate model in BO compared to GP ARD and RF.

The hypothesis that the lower $EF_{\text{max}}$ are caused by intrinsic acceleration and enhancement resulting from dataset collection process can be verified by collecting a subset from the uniformed grid sampled crossed barrel dataset. This subset is collected by running BO algorithm GP: EI until all candidates with top 5% toughness are found, representing an "easier" path towards optimums, and therefore carries intrinsic enhancement and acceleration. We run the same benchmarking framework on this subset, and observe that $EF_{\text{max}}$ is reduced, as seen in Figure S4 from Appendix. Despite the differences described above, all the investigated BO algorithms outperformed the random baseline demonstrating the efficacy of BO in materials optimization campaigns.
Chapter 4

Discussion

In this section, I further compare GP ARD, RF, and GP as surrogate models in BO under the context of autonomous and high-throughput materials optimization.

4.1 Comparison of RF and GP as surrogate models

While BO algorithms equipped with GP type surrogate models have been extensively used in many published materials studies, the observations above show observe that the performance of RF as surrogate model is can be comparable to that of GP. The results heavily suggest that RF is a capable surrogate model to consider besides GP in BO for future autonomous materials optimization campaigns.

Before more detailed comparison, an important distinction to make between RF and GP as surrogate models is how they predict mean and standard deviation, which can be attributed to the implicit assumptions when using them as surrogates to represent an unknown ground truth within materials domains. A GP in this work, whether with ARD or not, is essentially a distribution over a materials domain such that any finite selection of data points in this design space results in a multivariate Gaussian density over any point of interest in the space. For the selection of a new data point as next experiment, its predicted mean and standard deviation are all part of a gaussian distribution constructed from previous experiments. Therefore, the predicted means and standard deviations of GPs from their posteriors carry
gaussianity assumptions and can be interpreted as statistical predictions based on prior information. Meanwhile, a RF is an ensemble of decision trees, which are trained on the experimental data points collected during optimization and have slight variation due to bootstrapping. For RF, prediction of objective value at a new data point is an aggregated result, most likely the mean $\mu$ of all its decision trees’ prediction as used in this study; similarly, a prediction by RF of standard deviation at new data point is the standard deviation of all its decision trees’ predictions. Compared to those of GPs, the predicted means and standard deviations of RFs do not have strong distributional assumptions, and can be interpreted as empirical estimates. In short, during prediction, GPs rely on heavy distributional assumptions while RF is distribution free. Such difference between two surrogate models carries over to the values of mean and standard deviation, and together affect the selection of next experiment despite being paired with exact same acquisition function. This important distinction explains the differences of results in Figure 3 and Figure 4. Other factors affecting surrogate model selection are further discussed via the following side-by-side comparisons of GPs and RFs.

I would first like to discuss the time complexity of these surrogate models. Most commonly in autonomous experimental materials optimization studies, time spent on generating samples is much more significant than that of surrogate model training. However, in this study, there was a noticeable difference in time when benchmarking different BO algorithms, which reveals foreseeable challenges to future research. Across five datasets in this study, starting from the same initial experiments and using the same acquisition function $\text{LCB}_\tau$, the ratio of average running time to finish benchmarking framework between the three surrogate models is $t_{RF} : t_{GP} : t_{GP\text{ARD}} = 1 : 1.32 : 1.54$. With $n$ as the number of training data, $n_{\text{dim}}$ as design space dimension, $n_{\text{tree}}$ as number of decisions trees kept in RF model, in terms of general training time complexity, there is $t_{RF} = \mathcal{O}(n \log(n) \cdot n_{\text{dim}} \cdot n_{\text{tree}}) < t_{GP} = \mathcal{O}(n^3 + n^2 \cdot n_{\text{dim}})$. The relatively expensive computational complexity of GP model is mostly due to the process of calculating the inverse of an $n$ by $n$ matrix during its training process, and keeping anisotropic kernels certainly adds extra computational time. For reasonable choice of
$n$, $n_{\text{dim}}$, and $n_{\text{tree}}$ in physical science research, RF not only can be trained even faster via parallel computing of its decision trees, but also suffer less in performance with increasing $n$ than GP. While the $O(n^3)$ time complexity of GPs is typically less of a concern when working with smaller datasets, it could quickly become intractable when applied to larger datasets after experimental samples can be generated at unprecedented rate through advancement of automation in materials laboratories. At each learning cycle, time used in synthesis will eventually match with the time used in model retraining and prediction, typically within seconds. Therefore, in the future, if budget is no longer considered as number of experiments, but instead as total time invested, then RF has a potential advantage over GP when researchers aim to have fast and seamless feedback loop between model and materials experiments.

I would like to next discuss some properties in the RF, GP, and GP ARD models that would explain the observed differences in performance. The benchmarking effort presented here is unique in its use of abundant experimental materials data with built in noise, often unavoidable in physical science research. The results thus provide a realistic performance evaluation for optimization algorithms in the context of materials research. RF having good performance across the five experimental datasets can be partially attributed to predictions of RF being empirical estimates from its ensemble of decision trees and free of distributional assumptions. Each individual decision tree in RF naturally have low bias and high variance; the aggregation process in RF of different decision trees mitigates this issue, resulting in a model that has relatively low bias and medium variance, and thus are likely more robust to noise and applicable for generalized prediction with unknown assumption \[40,32\]. These properties of RF can be partially observed in Figure 4, where performance of RF match that of GP ARD as surrogate model. It can be also seen that the variance of $E_{F_{\text{max}}}$ for RF is on average lower than those of GP with ARD and GP without ARD. Generally speaking, on one hand, if the ground truth manifold of a design space indeed satisfies gaussianity assumption of GPs, then arguably GP type surrogate models have an advantage in learning a model with low bias and variance. On the other hand, if there were sharp discontinuities, piece-wise constants or changes in orders of magnitude through local
regions of materials design space, decision trees of RF would be able to capture these points accurately and reflect their influences on future predictions via aggregated result. These points are specifically regions of interest to be further investigated by researchers, whether they are new findings or outliers from experiments, but they typically fall out of the distributional assumptions of GP and are smoothed out in the learned model without considerable effort in kernel hyperparameter tuning.

I last discuss the effort required hyperparameter tuning of surrogate models during optimization. Despite HTE drastically increasing the rate of materials data collection, active learning for optimization in new materials domains still requires data collection in a sequential or batched manner. While RF has potentially more hyperparameters such as \( n_{\text{tree}} \), max depth, and max split to select, it is less penalized for sub-optimal choice of hyperparameters compared to GP. In this study, across five datasets, as long as sufficient \( n_{\text{tree}} \) were used in RF, its performance as surrogate model in BO algorithm has been consistently comparable to that of GP. Other hyperparameters of RF have had less of an impact on its performance. Meanwhile, besides the implicit distribution assumption of using a GP type surrogate model, a kernel (covariance function) of GP specifies a specific prior on the domain. Choosing a kernel that is incompatible with the domain manifold could significantly slow down optimization conversion due to loss of generalization, as seen in Figure S5 - S9 from Appendix, where BO algorithms with GP surrogate models are highly sensitive to kernel selection. For example, Matérn52 kernel analytically requires the fitted GP to be 2 times differentiable in the mean-square sense \([4]\), which can be difficult to verify for unknown materials design spaces. Selecting such a kernel could introduce extra domain manifold assumptions to an unfamiliar design space, as researchers often have limited data to make confident distribution assumptions of the domain at optimization onset. Instead of devoting nontrivial experimental budget to optimize the kernels of GP using adaptive kernels \([17]\), automating kernel selection \([18]\) or keeping a library of kernels available via online learning, RF is an easier off-the-shelf option that allows one to make fewer structural assumptions about unfamiliar materials domains. If a GP surrogate model is still preferred, a Multilayer Perceptron (MLP) kernel \([19]\) mimicking neural networks would
be suggested as it has comparable performance to other kernels (see Figure S5 - S9 from Appendix).

Admittedly, this benchmarking framework might have given RF a slight advantage by discretizing the materials domain through actively acquiring a new datapoint at each cycle and limiting the choice of next experiments within the pool of undiscovered datapoints. However, the crossed barrel dataset has a sampling density, size, and range within its design space sufficient to cover its manifold complexity. A drawback of RF is that it performs poorly in extrapolation beyond the search space covered by training data, yet in the context of materials optimization campaigns, this disadvantage can be mitigated by clever design of initial experiments, namely adapting sampling strategies like latin hypercube sampling (LHS). In this way, not only is the pseudo random nature of selecting initial experiments preserved but also the sampling stage also covers a wider range of data in each dimension so that RF surrogate model would not have to often extrapolate to completely unknown regions. Considering the benchmarking results and the comparisons above, while GP ARD has shown good robustness as surrogate model in BO across five materials domains, RF warrants future consideration as an alternative surrogate model. For RF, it is relatively easier to select of initial model hyperparameters, distribution assumption free, and when paired with the intuitive tuning of LCB’s weights to adjust exploration and exploitation, it forms an alternative BO algorithm suitable for general materials optimization campaigns at early stages.

4.2 Benefits of using GP with anisotropic kernels

As mentioned earlier, ARD allows us to utilize individual lengthscales for each input dimension $j$ in the kernel function of GP, which are subsequently optimized along learning cycles. These lengthscales in an anisotropic kernel provide a "weight" for measuring relative relevancy of each feature to predicting the objective, i.e. understanding the sensitivity of objective value to each input feature dimension. The reason GP without ARD shows worse performance is as follows: it will have a single lengthscale in an isotropic kernel as scaling hyperparameter, which is at odds with the fact that each
input feature has its distinct contribution to the objective. Depending on how different each feature is in nature, range and units, e.g. solvent composition vs. printing speed, using the same lengthscales in a kernel function for each feature dimension could provide unreliable predictive results. The materials optimization objective naturally has different sensitivities to each input variable, and thus it is rationale then, that the "lengthscale" parameter inside the GP kernel should be independent. In Figure 4, the noticeable improvements of using an anisotropic kernel can be seen in the relative lower performance of GP without ARD compared to that of GP with ARD. While data normalization can partially alleviate the problem, how it is conducted is highly subject to a researcher’s choice, and therefore I would like to raise awareness of the benefits of using GP with anisotropic kernels.

In addition, the lengthscales from the kernels of GP with ARD provides us with more useful information about the input features. These lengthscales values have been used for removing irrelevant inputs \[4\], where high \(l_j\) values imply low relevancy input feature \(j\). In the context of materials optimization, I find the following use of ARD especially useful: ARD could identify a few directions in the input space with specially high “relevance.” This means that if GP with ARD is trained on input data with their original units and without normalization, once the lengthscales of each feature \(l_j\) is extracted, the GP model in theory should not be able to accurately extrapolate more than \(l_j\) units away from collected observations in \(j^{th}\) dimension. Thus, \(l_j\) suggests the range of next experiments to be performed in the \(j^{th}\) dimension of the materials design space. It also infers a suitable sampling density in each dimension in the experimental setting. When a particular input feature dimension has a relative small \(l_j\) or large \(\frac{1}{l_j}\), it means that for small change in objective value, one would have a relatively large change in the location within this input feature dimension; thus, the sampling density or resolution in this dimension should be high enough to capture such sensitivity. In addition, people have considered using information extracted from these lengthscales for even more advanced analysis and variable selection \[50\]. At the expense of computation time tolerable in the context of materials optimization campaigns, an anisotropic kernel provides not only a better generalizable GP model
but also useful information in analyzing input feature relevancy at each learning cycle. For the above mentioned reasons, it would be great practice for researchers to emphasize their use of GP with anisotropic kernels as surrogate models during future materials optimization campaigns.
Chapter 5

Conclusions

In conclusion, I have benchmarked the performance of BO algorithms across five different experimental materials science domains. We utilize a pool-based active learning framework to approximate experimental materials optimization processes, and adapted active learning metrics to quantitatively evaluate the enhancement and acceleration of BO for common research objectives. We demonstrate that when paired with the same acquisition functions, RF as surrogate model can compete with GP with ARD, and both outperform GP without ARD. In the context of autonomous and high-throughput experimental materials research, I discussed the differences in implicit distributional assumptions of surrogate models and the benefits of using GP with anisotropic kernels. We provide practical insights on surrogate model selection for materials optimization campaigns, and also offer open source implementation of benchmarking code and datasets to support future algorithmic development.

Establishing benchmarks for active learning algorithms like BO across a broad scope of materials systems is only a starting point. Our observations demonstrate how the choice of active learning algorithms has to adapt to their applications in materials science, motivating more efficient ML guided closed-loop experimentation, and will likely directly result in a larger number of successful optimization of materials with record breaking properties. The impact of this work can be extended to not only other materials systems, but also a broader scope of scientific studies utilizing closed-loop and high-throughput research platforms. Through this benchmarking effort, I hope to
share more insights with the field of accelerated materials discovery and motivate a closer collaboration between ML and physical science communities.
Chapter 6

Appendix

This section provides supplementary figures for the study "Benchmarking the Performance of Bayesian Optimization across Multiple Experimental Materials Science Domains."

Table T1 - T5 describe the design space of the five experimental datasets in more detail.

Figure S1 shows the normalized RMSE values of RF models with different number of decision trees $n_{\text{tree}}$ during cross validation across five datasets.

Figure S2 show cross validation results of neural network, GP ARD, and RF ground truth models on Crossed barrel dataset and comparison of the effectiveness of BO algorithms when applied on each of the ground truth models. Figure S3 show cross validation results of neural network, GP ARD, and RF ground truth models on AgNP dataset and comparison of the effectiveness of BO algorithms when applied on each of the ground truth models. All ground truth models have noise $e(\mathbf{x}) = \mathcal{N}(0, 0.01\mu)$ at each point $\mathbf{x}$ in its design space, where $\mu$ is the mean of the dataset’s objective values. Neural network and RF model represent ground truth models that do not carry strong gaussianity assumption like GP. When ground truth model is GP type, it is observed that the relative optimization performance of BO algorithms with GP type surrogate models are improved over those with RF type surrogate models. GP type surrogate models' advantage over RF type ones is reduced when ground truth model is NN or RF, which are free of distributional assumptions. This confirms the
aforementioned concern that specific ground truth models could introduce extra bias into design space that could impact the benchmarking results. Nevertheless, GP with anisotropic kernels (GP ARD) is shown to be better surrogate model across all ground truth models, proving to be a robust model for future optimization campaigns.

Figure S4 shows the absolute $\text{EF}_{\text{max}}$ values shown by BO algorithms when guiding materials optimization campaigns across five datasets. It can be viewed together with Figure 4 in the manuscript, where $\text{EF}_{\text{max}}$ values are normalized for comparison.

Figure S5 - S9 show the performance of BO algorithms across five datasets. Compared to Figure 3 in manuscript, they are more comprehensive and include more surrogate models, acquisition functions, a range of $\bar{\lambda}$ values for LCB and more kernels for GPs.
Table T1: Crossed barrel dataset input feature space. The dataset has size 600. It consists of design parameters for the crossed barrel structure. The crossed barrel structures were optimized for max toughness.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Kind</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Discrete</td>
<td>[6, 12] with interval of 2</td>
<td>number of struts</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Discrete</td>
<td>[0, 200] with interval of 25</td>
<td>twisting angle [$^\circ$]</td>
</tr>
<tr>
<td>r</td>
<td>Discrete</td>
<td>[1.5, 2.5] with interval of 0.1</td>
<td>thickness [mm]</td>
</tr>
<tr>
<td>t</td>
<td>Discrete</td>
<td>[0.7, 1.4] with interval of 0.35</td>
<td>outer radius [mm]</td>
</tr>
</tbody>
</table>

Table T2: AgNP dataset input features space. The dataset has size 164. It consists of processing parameters for synthesizing triangular nanoprisms. The synthesized silver nanoparticles were optimized for shape and correspondingly absorbance spectrum. $Q_i$ is the ratio between flow rate of reactant $i$ to total aqueous flow rate.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Kind</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{\text{seed}}$</td>
<td>Discrete</td>
<td>[0.5:80] with interval of 5</td>
<td>flow rate ratio of Ag seeds [%]</td>
</tr>
<tr>
<td>$Q_{\text{AgNO}_3}$</td>
<td>Discrete</td>
<td>[0.5:80] with interval of 5</td>
<td>flow rate ratio of silver nitrate [%]</td>
</tr>
<tr>
<td>$Q_{\text{TSC}}$</td>
<td>Discrete</td>
<td>[0.5:80] with interval of 5</td>
<td>flow rate ratio of trisodium citrate [%]</td>
</tr>
<tr>
<td>$Q_{\text{PVA}}$</td>
<td>Discrete</td>
<td>[10:40] with interval of 5</td>
<td>flow rate ratio of polyvinyl alcohol [%]</td>
</tr>
<tr>
<td>$Q_{\text{total}}$</td>
<td>Discrete</td>
<td>[200:1000] with interval of 100</td>
<td>total flow rate [µL/min]</td>
</tr>
</tbody>
</table>

Table T3: P3HT/CNT dataset input features space. The dataset has size 178. It consists of composition parameters for carbon nanotube polymer blend. The composite blends were optimized for electrical conductivity [S/cm]. A constraint on the parameter space is $P3HT + D_1 + D_2 + D_6 + D_8 = 1$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Kind</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P3HT</td>
<td>Continuous</td>
<td>[0, 100]</td>
<td>composition ratio of P3HT [%]</td>
</tr>
<tr>
<td>D_1</td>
<td>Continuous</td>
<td>[0, 100]</td>
<td>composition ratio of D1 CNT sample [%]</td>
</tr>
<tr>
<td>D_2</td>
<td>Continuous</td>
<td>[0, 100]</td>
<td>composition ratio of D2 CNT sample [%]</td>
</tr>
<tr>
<td>D_6</td>
<td>Continuous</td>
<td>[0, 100]</td>
<td>composition ratio of D6 CNT sample [%]</td>
</tr>
<tr>
<td>D_8</td>
<td>Continuous</td>
<td>[0, 100]</td>
<td>composition ratio of D8 CNT sample [%]</td>
</tr>
</tbody>
</table>

Table T4: Perovskite dataset input features space. The dataset has size 94. It consists of composition parameters for halide perovskites $\text{Cs}_x\text{MA}_y\text{FA}_{1-x-y}\text{PbI}_3$ thin films. The perovskite films were optimized for environmental stability. A constraint on the parameter space is $\text{CsPbI} + \text{FAPbI} + \text{MAPbI} = 1$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Kind</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CsPbI</td>
<td>Discrete</td>
<td>[0, 100] with interval of 1</td>
<td>composition ratio of CsPbI [%]</td>
</tr>
<tr>
<td>FAPbI</td>
<td>Discrete</td>
<td>[0, 100] with interval of 1</td>
<td>composition ratio of FAPbI [%]</td>
</tr>
<tr>
<td>MAPbI</td>
<td>Discrete</td>
<td>[0, 100] with interval of 1</td>
<td>composition ratio of MAPbI [%]</td>
</tr>
</tbody>
</table>

Table T5: AutoAM dataset input features space. The dataset has size 100. It consists of additive manufacturing control parameters for printing specific shapes. The additive manufacturing system was optimized for printing shapes with best shape score.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Kind</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prime Delay</td>
<td>Continuous</td>
<td>[0, 5]</td>
<td>delay in extrusion before deposition [s]</td>
</tr>
<tr>
<td>Print Speed</td>
<td>Continuous</td>
<td>[0.1, 10]</td>
<td>speed of printing movement [mm/s]</td>
</tr>
<tr>
<td>X Offset Correction</td>
<td>Continuous</td>
<td>[-1, 1]</td>
<td>printing head offset in X direction [mm]</td>
</tr>
<tr>
<td>Y Offset Correction</td>
<td>Continuous</td>
<td>[-1, 1]</td>
<td>printing head offset in Y direction [mm]</td>
</tr>
</tbody>
</table>
Figure S1: The normalized RMSE values of RF models with different number of decision trees $n_{\text{tree}}$ during cross validation. (a)(b)(c)(d)(e) are results for Crossed barrel, AgNP, P3HT, Perovskite, and AutoAM dataset respectively. The RF models had $n_{\text{tree}}$ between 5 to 300. For each model, 50 different random test and train split were conducted, followed by 5-fold cross validation and evaluation of model on the respective training set and test set. Variation of normalized RMSE at each $n_{\text{tree}}$ is visualized by plotting the median as well as shaded regions representing the 5th to 95th percentile of the 50 different evaluations. The figure shows that RF with $n_{\text{tree}} = 100$ is a suitable initial hyperparameter for RF surrogate models because it not only achieves similar prediction accuracy as RF models with larger $n_{\text{tree}}$ but also avoids the risk of overfitting from larger $n_{\text{tree}}$. 
Figure S2: The Leave-one-out cross validation results shown in parity plots for neural network, GP, and RF models fitted on the Crossed barrel dataset and the performance of different BO algorithms on these emulated design spaces. In (a)(d), the neural network model has structure (4, 120, 240, 480, 960, 480, 240, 120, 1). It is fully connected between layers and has activation function Leaky ReLU. In (b)(e), the GP model is anisotropic and uses Mátern52 kernel. In (c)(f), the RF model has an ensemble of 500 decision trees and is trained with bootstrapping. BO algorithms are trained for 30 times as they start from different initial experiment locations. Variation at each learning cycle $i$ is visualized by plotting the median as well as shaded regions representing the 5th to 95th percentile of the aggregated 30-run ensembles.
Figure S3: The Leave-one-out cross validation results shown in parity plots for neural network, GP, and RF models fitted on the AgNP dataset and the performance of different BO algorithms on these emulated design spaces. In (a)(d), the neural network model has structure (5, 60, 120, 240, 480, 240, 120, 60, 1). It is fully connected between layers and has activation function Leaky ReLU. In (b)(e), the GP model is anisotropic and uses Mátern52 kernel. In (c)(f), the RF model has an ensemble of 500 decision trees and is trained with bootstrapping. BO algorithms are trained for 30 times as they start from different initial experiment locations. Variation at each learning cycle $i$ is visualized by plotting the median as well as shaded regions representing the 5$\text{th}$ to 95$\text{th}$ percentile of the aggregated 30-run ensembles.
Figure S4: Absolute EF_{max} demonstrated by BO algorithms equipping GP without ARD, GP with ARD, and RF as surrogate models and all using LCB_{θ} as acquisition function. For each algorithm applied across datasets, the median of EF_{max} is shown by the barplots, and its 5^{th} and 95^{th} percentile are shown by respective floating bars. The crossed barrel subset is collected by running BO algorithm GP: EI until all candidates with top 5% toughness are found, representing an "easier" path towards optimums.
Figure S5: The aggregated performance of BO algorithms on the Crossed barrel dataset. The comprehensive benchmark involves multiple surrogate models, kernel type for GP, and acquisition functions. The performance of BO algorithms with GP ARD, GP, and RF as surrogate model can be observed in (a)(d)(g), (b)(e)(h), and (c)(f)(i) respectively. Variation at each learning cycle is visualized by plotting the median as well as shaded regions representing the 5th to 95th percentile of the aggregated 50-run ensembles.
Figure S6: The aggregated performance of BO algorithms on the AgNP dataset. The comprehensive benchmark involves multiple surrogate models, kernel type for GP, and acquisition functions. The performance of BO algorithms with GP ARD surrogate model and various kernels and acquisition functions can be observed in (a)(e)(i) and (b)(f)(j). The performance of BO algorithms with GP surrogate model and various kernels and acquisition functions can be observed in (c)(g)(k). The performance of BO algorithms with RF surrogate model and various kernels and acquisition functions can be observed in (d)(h)(l). Variation at each learning cycle is visualized by plotting the median as well as shaded regions representing the 5$^{th}$ to 95$^{th}$ percentile of the aggregated 50-run ensembles.
Figure S7: The aggregated performance of BO algorithms on the AutoAM dataset. The comprehensive benchmark involves multiple surrogate models, kernel type for GP, and acquisition functions. The performance of BO algorithms with GP ARD surrogate model and various kernels and acquisition functions can be observed in (a)(e)(i) and (b)(f)(j). The performance of BO algorithms with GP surrogate model and various kernels and acquisition functions can be observed in (c)(g)(k). The performance of BO algorithms with RF surrogate model and various kernels and acquisition functions can be observed in (d)(h)(l). Variation at each learning cycle is visualized by plotting the median as well as shaded regions representing the 5th to 95th percentile of the aggregated 50-run ensembles.
Figure S8: The aggregated performance of BO algorithms on the P3HT/CNT dataset. The comprehensive benchmark involves multiple surrogate models, kernel type for GP, and acquisition functions. The performance of BO algorithms with GP ARD surrogate model and various kernels and acquisition functions can be observed in (a)(e)(i) and (b)(f)(j). The performance of BO algorithms with GP surrogate model and various kernels and acquisition functions can be observed in (c)(g)(k). The performance of BO algorithms with RF surrogate model and various kernels and acquisition functions can be observed in (d)(h)(l). Variation at each learning cycle is visualized by plotting the median as well as shaded regions representing the 5th to 95th percentile of the aggregated 50-run ensembles.
Figure S9: The aggregated performance of BO algorithms on the Perovskite dataset. The comprehensive benchmark involves multiple surrogate models, kernel type for GP, and acquisition functions. The performance of BO algorithms with GP ARD, GP, and RF as surrogate model can be observed in (a)(d)(g), (b)(e)(h), and (c)(f)(i) respectively. Variation at each learning cycle is visualized by plotting the median as well as shaded regions representing the 5th to 95th percentile of the aggregated 50-run ensembles.
References


[50] Topi Paananen, Juho Piironen, Michael Riis Andersen, and Aki Vehtari. Variable selection for gaussian processes via sensitivity analysis of the posterior predictive