NASA/TM-20260001102

Benchmarking Bayesian Optimization Frameworks and Acquisition Strategies for Materials Discovery and Autonomous Laboratories

Joshua Stuckner and Peter Toma
Glenn Research Center, Cleveland, Ohio

Jacob Goodin
The University of Akron, Akron, Ohio

Brandon Hearley
Glenn Research Center, Cleveland, Ohio

Stephen Xie
KBR, Inc., Moffett Field, California

March 2026

NASA STI Program Report Series

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION.
  Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM.
  Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT.
  Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION.
  Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION.
  Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION.
  English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

NASA/TM-20260001102

Benchmarking Bayesian Optimization Frameworks and Acquisition Strategies for Materials Discovery and Autonomous Laboratories

Joshua Stuckner and Peter Toma
Glenn Research Center, Cleveland, Ohio

Jacob Goodin
The University of Akron, Akron, Ohio

Brandon Hearley
Glenn Research Center, Cleveland, Ohio

Stephen Xie
KBR, Inc., Moffett Field, California

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

March 2026

This work was sponsored by the
Transformative Aeronautics Concepts Program.

Level of Review: This material has been technically reviewed by technical management.

This report is available in electronic form at https://www.sti.nasa.gov/ and https://ntrs.nasa.gov/

NASA STI Program/Mail Stop 050
NASA Langley Research Center
Hampton, VA 23681-2199

Benchmarking Bayesian Optimization Frameworks and Acquisition Strategies for Materials Discovery and Autonomous Laboratories

Joshua Stuckner and Peter Toma*
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Jacob Goodin
University of Akron
Akron, Ohio 44325

Brandon Hearley
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Stephen Xie
KBR, Inc.
Moffett Field, California 94035

Abstract

Bayesian optimization (BO) can accelerate materials discovery by guiding expensive experiments toward the most promising processing conditions. We systematically compare five BO surrogate and framework combinations (Gaussian processes in Ax, Gaussian processes and Monte-Carlo neural networks in BayBE, random forests in Lolopy, and tree-structured Parzen (TPE) estimators in Hyperopt) on three benchmarks that mimic common materials design tasks (a discrete solid-electrolyte composition space, a hybrid discrete/continuous laminate-composite design problem solved with micromechanics modeling, and the continuous Ishigami analytic function which is a standard optimization benchmark). Each BO surrogate is paired with posterior mean, probability of improvement, and expected improvement acquisition functions and run for 100 trials from randomized initial samples with uniform random search providing a control. Across five random seeds per setting, BayBE's Gaussian-process surrogate with expected improvement consistently reached ≥95 % of the known optimum in the fewest evaluations, while Lolopy's random forest matched or exceeded GP performance on purely categorical or mixed spaces at a higher computational cost. Posterior mean alone often stagnated at local optima, underscoring the need for exploration, whereas probability and expected improvement balanced exploration and exploitation leading to better optimization in fewer trials. Execution times ranged from milliseconds for TPE to minutes for neural-network and random-forest surrogates. These results establish baseline expectations for BO in automated materials laboratories and highlight expected improvement with Gaussian processes as a reliable first choice, with random forests offering a strong alternative when categorical variables dominate. The benchmark suite and code are released to facilitate future surrogate, acquisition, and constraint-handling research in data-driven materials optimization.

*NASA Office of STEM Engagement Spring 2024 Intern, University of Florida, undergraduate.

Introduction

Discovering better materials still relies heavily on labor-intensive trial-and-error experimentation, limiting the pace at which better batteries, lighter aircraft components, and other advanced technologies reach society. Autonomous laboratories that pair rapid synthesis and characterization tools with machine-learning algorithms promise to accelerate this search, but their real-world impact depends on how efficiently those algorithms decide which experiment to perform next. Bayesian optimization (BO), which iteratively balances trying uncertain conditions against refining the best-known recipe, has emerged as a leading approach, yet the community lacks clear guidance on which surrogate models, acquisition functions and software frameworks work best across the mixed continuous-and-categorical search spaces common in materials design [1]. Here we show, using three representative benchmarks that span purely continuous, purely categorical, and hybrid design spaces, that a Gaussian process (GP) surrogate implemented in the BayBE framework and paired with an expected-improvement acquisition function consistently reaches at least 95% of each benchmark's optimum in the fewest experimental evaluations. Random-forest surrogates equal or surpass Gaussian processes on categorical tasks. Posterior-mean acquisition frequently performs the worst of the tested acquisition functions. Neural network (NN) and tree-structured Parzen estimators (TPE), often promoted for high-dimensional problems, underperform in the low-data regimes typical of early-stage materials development campaigns. By establishing quantitative baselines and exposing strengths and weaknesses for optimizing discrete chemistries, composite materials, and nonlinear analytic landscapes, our study provides practical rules of thumb for configuring BO in automated materials laboratories. These findings will aid researchers in allocating scarce experimental budgets more effectively and furnish a benchmark suite for testing next-generation surrogate models, constraint handlers, and multi-objective strategies aimed at autonomous discovery platforms.

BO is an iterative process with five general steps illustrated in Figure 1. 1. The process begins with some initial data. This data could be pre-existing or sampled at the start. The purpose of this data is to provide a foundation for the surrogate model. 2. A surrogate model is trained using the initial data. This model is designed to predict the output of the source function or experimental process (referred to as the 'Source') and quantify the uncertainty associated with these predictions. 3. An optimizer is used to find the input that maximizes an acquisition function derived from the surrogate model. The acquisition function can be chosen to balance exploration (sampling where the uncertainty is high) and exploitation (sampling where the surrogate model predicts high performance). 4. The Source is sampled at the input point that optimizes the acquisition function. The result of this sampling is then added to the initial training data. 5. Steps 2 to 4 are repeated until a desired outcome is achieved, or the resources allocated for the experiment are exhausted.

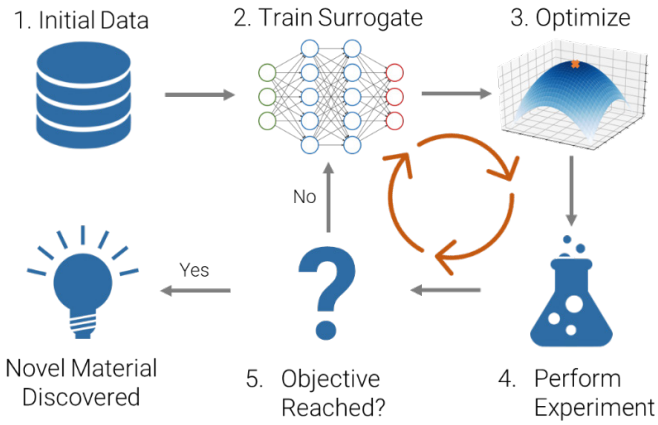


Figure 1.—Bayesian optimization process.

There has been a recent surge in research in Bayesian optimization for materials due in part by the increasing availability of open-source libraries that implement state-of-the-art BO techniques. Many frameworks and tools have been introduced in the last several years. We highlight many impactful tools here, although we did not evaluate all of them. BoTorch is a PyTorch based-library that provides a modular, differentiable framework for BO [2]. Ax, first released in 2019, is a higher-level platform that builds on top of BoTorch and simplifies experimental optimization [3]. GPyOpt was one of the original Python libraries (2016) for GP-based BO and supports parallel evaluation, various kernels, and user-defined constraints. The BayesianOptimization package by Fernando Nogueira (often just called bayes_opt) is another lightweight library that became popular for quick BO tasks; it uses Gaussian processes (via GPy or Sklearn) under the hood and provides a simple interface. Hyperopt, Optuna, and SMAC (Sequential Model-based Algorithm Configuration) were developed originally for machine learning hyperparameter tuning. Hyperopt uses the TPE method (a form of sequential model-based optimization) and can run trials in parallel or serial [4]. Optuna (introduced around 2018) uses an adaptive TPE and offers a very user-friendly interface for defining search spaces, including conditional parameters [5]. Lolo and SMAC3 use random forest surrogate models [6], [7]. DragonFly implements methods for higher dimensional domains and for handling multifidelity evaluations when cheap approximations of an expensive function are available [8]. COMBO (combinatorial materials optimization) has been applied to problems like discovering new molecules and crystals and implements Thompson sampling [9]. Summit was designed for optimizing chemical processes [10]. Olympus offers benchmarks and algorithms commonly used for noisy experimental optimization, especially for materials and chemistry [11]. Atlas builds on Olympus and is designed with robotic laboratories in mind [12]. Experimental Design via Bayesian Optimization (EDBO) is a practical implementation of Bayesian optimization for chemical synthesis [13]. GAUssian processes in CHEmistry (GAUCHE) provides Gaussian kernels defined over structured inputs such as graphs and strings for better application to molecules, proteins, and chemical reactions [14].

There have been many recent successful applications of BO in materials design and discovery using increasingly sophisticated techniques. Schmidt et al. showed how BO can drive closed-loop robotic experimentation while monitoring data quality [15]. Xian et al. combined reinforcement learning with early-stage BO exploration to accelerate black-box materials design [16]. BO was used to steer density functional theory (DFT) calculations to optimize magnetocrystalline anisotropy [17]. Sabanza-Gil et al. performed a systematic study of the cost/accuracy trade-offs when using Multifidelity BO in materials and molecular research [18]. Multifidelity BO has also been used recently to accelerate the automated discovery of drug molecules [19] and perform high throughput materials screening [20]. Multi-objective BO is becoming a popular choice to co-optimize for multiple material properties including applications in material extrusion [21], high temperature alloys [22], and turbine engine blade alloys [23]. Techniques are even being used to efficiently map entire pareto fronts of material properties so that material selection can be based on application specific property trade-offs [24][25]. The processing-structure-property relationships critical for designing new materials are very complex, often non-linear, and high dimensional. Advanced surrogate models are being applied to capture these relationships more efficiently within BO frameworks including neural networks [26], physics informed kernels [27], and hierarchical GPs [28].

Our study directly compares BO frameworks using GP, random forest, and neural network surrogates combined with several different acquisition functions on continuous, categorical, and hybrid materials benchmarks. We provide quantitative comparisons to select BO strategies for next-generation autonomous laboratories and make our benchmarks publicly available for the evaluation of future BO advancements on materials science problems.

Methods

Many experiments were conducted to compare different BO methods. Each experiment selected from a choice of data sources (i.e., benchmark problems), surrogates to model the data, acquisition functions, optimization methods, and software frameworks. Not every combination of choices was compatible; for example, many surrogates required certain optimization methods or software frameworks as detailed in the following section. For each benchmark, a uniform random search was used as a control for comparison.

Surrogates

Four types of surrogate models were compared for performance: Gaussian processes (GP), Monte Carlo neural networks using dropout [29], random forests [30], [31], and TPE [32], [33].

A GP approximates a distribution of observations as a multivariate Gaussian distribution:

$$\mathbf{y} \sim N\big(m(\mathbf{x}), k(\mathbf{x},\mathbf{x})\big) \tag{1}$$

where **x** and **y** are the training inputs and outputs, respectively, *m* is the mean function, and *k* is the covariance or Gram matrix. On this distribution, the estimated mean and variance of a test input x^* are described by the equations:

$$\mu(x^*) = m(x^*) + \mathbf{k}(x^*,\mathbf{x})^{\mathrm{T}}(k + \sigma_n^2 I)^{-1}(\mathbf{y} - m(\mathbf{x})) \tag{2}$$

$$\sigma(x^*) = k(x^*,x^*) - \mathbf{k}(x^*,\mathbf{x})^{\mathrm{T}}(k + \sigma_n^2 I)^{-1}\mathbf{k}(x^*,\mathbf{x}) \tag{3}$$

where σ_n^2 is the variance hyperparameter and $\mathbf{k}(x^*,\mathbf{x})$ is the variance between x^* and each entry in **x**:

$$\mathbf{k}(x^*,\mathbf{x}) = [k(x^*,x_1),\ldots,k(x^*,x_n)]^{\mathrm{T}} \tag{4}$$

Two different Bayesian Optimization (BO) frameworks using GPs were compared: the Adaptive Experimentation Platform (Ax) [3], and Bayesian Back End (BayBE) [34], an alternate framework developed for use in materials discovery and related applications. For both frameworks, the GP surrogates utilized the BoTorch [2] SingleTaskGP as an internal model with exact marginal log-likelihood as a loss function, a trainable constant mean, and a Matérn kernel [35] for covariance. Training used GPyTorch's [36] Blackbox Matrix-Matrix multiplication method. On the solid electrolyte benchmark, Ax did not support the categorical constraints of the problem, although this did not significantly impact optimization.

A NN architecture implemented in PyTorch was used for the Monte Carlo NN surrogates using dropout. The NN models used a fully connected input layer mapping the input vector to a constant number of internal neurons, followed by a variable number of hidden layers and then a final output layer with a single output value. Except for the final layer, each linear layer was followed by batch normalization, SiLU [37] activation, and finally dropout layers. Unlike most networks using dropout, the dropout layers remained enabled during the surrogate's posterior evaluation and the network was called repeatedly on the evaluation data to provide the posterior mean and standard deviation of the data, with a dropout rate of 0.05 chosen by experimentation to provide good accuracy while still producing a useful variance result. The network was reinitialized with a constant random seed for every BO step.

To compensate for the low quantity of training data (of less than 100 data points), a multistep process was used for neural network training. During each training step, the previously sampled dataset was divided into a training set and test set with an 80/20 train-test split. The NN was trained for up to

100 epochs, with early stopping if the validation loss did not improve for a number of epochs above a "patience" constant. The epoch number with the lowest validation loss was subsequently chosen as the ideal epoch count, and the network was reset and trained on the full dataset for that number of epochs. Additionally, the NN hyperparameters were reoptimized every 10 BO iterations. An optimal learning rate, number of internal neurons, and number of hidden layers were chosen by finding the combination with the lowest validation loss through a grid search. Hyperparameter tuning could be accelerated through BO, but for this work with relatively few samples, model training was quick enough that grid search was tractable. The possible learning rates were 0.01, 0.005, 0.001, 0.0005, and 0.0001. The possible internal neuron counts were 16, 64, 256, and 1024, and the possible number of hidden layers were 2, 4, and 6.

The random forest model used the Lolopy framework, which has been shown to significantly reduce the number of experiments required to achieve target properties compared to random search [7]. Uncertainty predictions were determined by standard deviation of predictions from an ensemble of random forest models. Random forests are built from an ensemble of decision trees which each contribute to a final prediction. Each decision tree is trained on random subsets of the training data, a technique known as bootstrap aggregating or bagging [31]. To make a prediction, the random forest aggregates the predictions of all its decision trees, typically by taking the majority vote for classification tasks or averaging the predictions for regression tasks [30]. Each decision tree makes predictions by following a series of binary decisions based on input features. These decisions split the data into different groupings to achieve the desired model output. The binary decision at each branch is determined by selecting a single input feature and a numerical threshold that best splits the data. The quality of a split is evaluated using metrics such as the Gini impurity or information gain. The Gini impurity measures how often a randomly chosen element would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. The decision at each branch aims to split the data as purely as possible, minimizing the Gini impurity of the resulting subsets. Random forests are popular because they tend to produce highly accurate predictions, are less prone to overfitting compared to individual decision trees and can handle large datasets with higher dimensionality effectively. However, random forests and other tree-based models are incapable of extrapolation which is important when attempting to discover superior novel materials. As a BayBE surrogate requires a Torch gradient trace during posterior evaluation, the random forest model could not be used with BayBE and a lower-level Scipy-based optimization was used instead. For continuous variables, Scipy's *optimize.differential_evolution* function was used to find the input on the surrogate with the optimum acquisition function value, whereas for categorical variables, the optimizer conducted a grid search with every possible combination of categorical variables. These approaches were combined for hybrid search spaces, which ran Scipy's optimizer on every possible combination of categorical values to find the optimum acquisition function value.

The TPE models the objective function using two probability density functions: one for good observations and another for the rest. It samples new candidate points based on these densities to efficiently explore the search space [32]. TPE is especially effective for high-dimensional, complex spaces, often used in materials discovery for optimizing experimental parameters [38]. For the TPE surrogate, we used the adaptive [39] variant implemented in the Hyperopt [4] framework.

Acquisition Functions

Each surrogate model was tested using three different acquisition functions: posterior mean (PM), probability of improvement (PI), and expected improvement (EI).

The PM acquisition function is simply the predicted posterior mean of the surrogate,

$$\mathrm{PM}(x) = \mu(x) \tag{5}$$

The PI acquisition function is the probability that the input will match or surpass the previous optimum value,

$$\mathrm{PI}(x) = \boldsymbol{\Phi}\left(\frac{\mu(x)-f(x^*)}{\sigma(x)}\right) \tag{6}$$

where $\boldsymbol{\Phi}(z)$ is the cumulative distribution function, $f(x^*)$ is the previously observed optimum, and $\sigma(x)$ is the uncertainty.

Finally, the EI acquisition function attempts to maximize the expected value of the improvement on the previous optimum value at an input, according to the formula:

$$\mathrm{EI}(x) = \big(\mu(x) - f(x^*)\big)\boldsymbol{\Phi}\left(\frac{\mu(x)-f(x^*)}{\sigma(x)}\right) + \sigma(x)\varphi\left(\frac{\mu(x)-f(x^*)}{\sigma(x)}\right) \tag{7}$$

where $\varphi(z)$ is the probability density function of a normal distribution. PM is a completely greedy acquisition function, while PI and EI strike a balance between greed and exploration, where EI is somewhat greedier than PI. Greedy acquisition functions prioritize sampling where the model predicts the best outcome (exploitation), while exploratory acquisition functions balance sampling in uncertain regions to improve the model (exploration). The maximum uncertainty acquisition function could be chosen to optimize for exploration.

Benchmarks

Multiple benchmarks were selected to evaluate the performance of the tested surrogate models over continuous, discrete, and hybrid search spaces.

Ishigami Function

A purely continuous search space was represented by the Ishigami function [40] benchmark, described by the following formula:

$$f(x) = \sin(x_1) + a\sin^2(x_2) + bx_3^4 \sin(x_1) \tag{8}$$

where x_1, x_2, and x_3 are input parameters and a and b are constants set to 7 and 0.1 respectively. The Ishigami function was chosen due to its strong nonlinearity and nonmonotonicity, representing a good challenge for Bayesian optimizers. An example response surface of the Ishigami function with two inputs is shown in Figure 2.

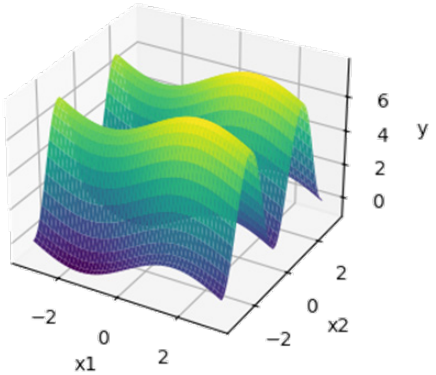


Figure 2.—Ishigami function with two input variables. The function is hard to optimize with local minima.

Solid Electrolyte

All-solid-state batteries are promising alternatives to liquid-based lithium- (Li) ion batteries that offer higher safety and energy density [41]. One crucial component of all-solid-state batteries is the solid electrolyte, through which Li ions migrate during charging and discharging [42]. While computational screening efforts have already identified many promising candidates, the breadth of the remaining search space is an ongoing challenge. For example, one way to increase the ionic conductivity of a solid electrolyte is to dope it with impurity elements [43]. However, the computational cost of predicting the doping effects depends on the number of elements considered as well as the number of possible crystal sites each may occupy. We developed a benchmark dataset to compare the efficiency of algorithms for exploring the effects of doping an ordered crystal structure.

Lithium argyrodite Li_6PS_5Cl, a solid electrolyte known for its high ionic conductivity, was selected as the representative ordered crystal structure. We defined the objective function for a given doped crystal structure as the sum of computed stability and transport metrics. The search space of the doping benchmarking problem is a five-dimensional problem with five to seven discrete values per dimension. The first four dimensions encode doping with aluminum (Al), oxygen (O), chlorine (Cl), and fluorine (F) elements. A null value indicates no doping and a positive integer indicates which crystallographic site is substituted. Al can replace one of six Li sites [44]. O can replace one of five sulfur (S) sites [45]. Excess Cl, via doping, can replace an S-site [44] whereas F can replace an S-site or the Cl site [46]. The fifth dimension encodes doping with antimony (Sb), silicon (Si), germanium (Ge), or tin (Sn) on the phosphorus (P) site [47]. A null value indicates no doping and a positive integer indicates which element is selected for substitution. We eliminate samples for which the same crystallographic site is selected for multiple dopants, resulting in 5,845 valid points in the search space.

The stability metric in this work is the distance from the convex hull of formation energies, computed using the M3GNet universal machine learning potential [48]. The transport metric is the 1-D migration barrier height for Li diffusion calculated through the bond valence site energy method [49], [50]. In practical searches, the objective function could include higher-fidelity property calculations using density functional theory.

Composite Materials

Composite materials are used in a wide variety of high-performance engineering applications due to their high strength to weight ratios and the ability to tailor the design of the material to a given application. Such materials are, however, subject to unique and complex failure modes that occur at multiple length scales due to the presence of different constituent materials (i.e., the fiber and the matrix) and their interface interactions. Thus, to truly leverage the benefits composite materials offer, validated multiscale models that can capture the mechanics of such materials at each length scale paired with advanced optimization techniques is necessary to design ‘fit-for-purpose’ materials. One such tool developed for analysis of composite materials is the Micromechanics Analysis Code with Generalized Method of Cells (MAC/GMC) [51]. The tool offers a suite of micromechanics theories to solve composite behavior, including the Generalized Method of Cells (GMC), first developed by Paley and Aboudi [52] and the High Fidelity Generalized Method of Cells, developed by Aboudi et al. [53]. Due to the semi-analytical nature of these theories, the tool can rapidly solve complex composite behavior at multiple length scales through the built-in homogenization and localization techniques, providing both global and local stress and strain fields on the order of seconds. Thus, the tool is an excellent candidate to pair with the inverse design framework to determine the optimal composite configuration for specified load cases.

The objective of the benchmark problem is to select the best laminate configuration that meets a defined safety criterion and minimizes the laminate weight. To limit the design space, the laminate is

assumed to be fixed to 8 total plies and symmetric, and the microstructure of each lamina is assumed to be hexagonally packed (Figure 3). The GMC micromechanics theory is employed in MAC/GMC for evaluation of a candidate laminate.

Given the assumptions made for the benchmark problem, model inputs to be optimized are the selection of constituent materials, the volume fraction, and the 8 ply orientations. The ply orientation angles are bounded between –90° and 90°, and the volume fraction is bounded between 0.3 and 0.7 to reflect realistic volume fractions in polymer matrix composites [54]. Further, the laminate is assumed to be symmetric, and thus only 4 ply angles need to be specified. The fiber and matrix materials are selected from a database of materials, with the properties to run MAC/GMC and evaluate the composite design defined in Table 1. Note that all fibers are assumed to be transversely isotropic, and all matrix materials are assumed to be isotropic. An isotropic fiber can be simulated by entering the same value for any property with subscripts of *L* and *T* (longitudinal and transverse material directions, respectively). Thus, a candidate laminate can be evaluated as a function of 7 inputs (Figure 4): the 4 ply orientation angles, the volume fraction, and the choice of one of eight fiber materials and four matrix materials from the database.

MAC/GMC provides the nonlinear stress-strain behavior for a composite given applied loading, including predictions in damage progression in the composite given a supplied failure criterion. For this benchmark problem, the Tsai-Hill failure criterion is implemented at the microscale by degrading the stiffness of each constituent locally based on the developed local stress fields in each subcell. MAC/GMC does not, however, output the evaluation of the Tsai-Hill criterion for each lamina in the composite, which is typically used in structural engineering to evaluate material design, and thus must be calculated externally with each MAC/GMC run. The Tsai-Hill criterion is defined as

$$1 \geq \left(\frac{\sigma_{11}}{X_{11}}\right)^2 - \left(\frac{\sigma_{11}\sigma_{22}}{X_{11}^2}\right) + \left(\frac{\sigma_{22}}{X_{22}}\right)^2 + \left(\frac{\tau_{12}}{X_{12}}\right)^2 \tag{9}$$

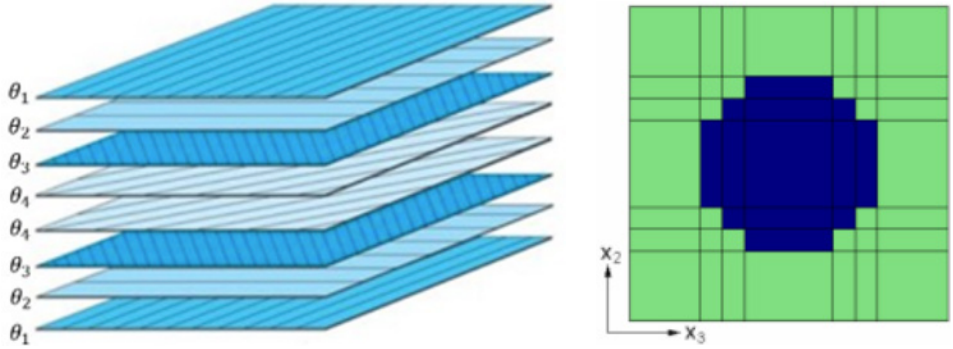
where σ_{ij} and τ_{ij} are the global normal and shear stresses, respectively, and X_{ij} are the stress allowables for a composite, and a value of one for the right-hand sign indicates the onset of failure. MAC/GMC evaluates the criterion at the microscale, and thus the composite allowables are not known. To solve for them, three additional analyses are needed with each evaluation of the load case:

1. Apply σ_{11} until failure to a single ply orientated at 0° and keep σ_{22} and τ_{12} constant at 0 stress. At the point of failure, Equation (9) reduces to $\sigma_{11} = X_{11}$
2. Apply σ_{22} until failure to a single ply orientated at 0° and keep σ_{11} and τ_{12} constant at 0 stress. At the point of failure, Equation (9) reduces to $\sigma_{22} = X_{22}$
3. Apply τ_{12} until failure to a single ply orientated at 0° and keep σ_{11} and σ_{22} constant at 0 stress. At the point of failure, Equation (9) reduces to $\tau_{12} = X_{12}$

With the lamina level allowables defined, the Tsai-Hill criterion can be evaluated for each ply from the MAC/GMC output for the full composite subject to the test load case. MAC/GMC outputs the stresses in each ply in the global coordinate system, and thus the solution stresses must be rotated to the local coordinate system to evaluate the failure criterion. The ply level stresses can be rotated using

$$\begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \tau_{12}\end{Bmatrix} = \begin{bmatrix}\cos^2\theta & \sin^2\theta & 2\cos\theta sin\theta\\ \sin^2\theta & \cos^2\theta & -2\cos\theta sin\theta\\ 2\cos\theta sin\theta & -2\cos\theta sin\theta & \cos^2\theta - \sin^2\theta\end{bmatrix}\begin{Bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \tau_{xy}\end{Bmatrix} \tag{10}$$

where θ is the orientation angle of the ply. With the stresses in the local coordinate system, the Tsai-Hill value, denoted herein as *TH*, can be calculated using the right hand side of Equation (9), which must be less than or equal to 1 for a safe design.

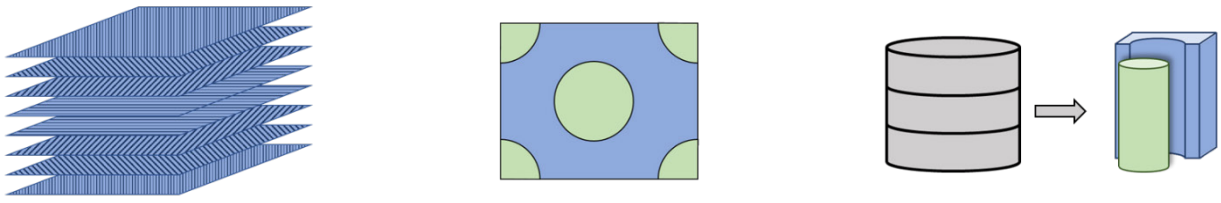


(a) (b)

Figure 3.—(a) Laminate configuration for an 8-ply symmetric composite and (b) Lamina microstructure in MAC/GMC, where blue subcells have fiber material properties and green subcells have matrix material properties

TABLE 1.—FIBER AND MATRIX PROPERTIES

Fiber	
Property	Symbol
Longitudinal modulus	E_L^f
Transverse modulus	E_T^f
Longitudinal Poisson's ratio	ν_L^f
Transverse Poisson's ratio	ν_T^f
Shear modulus	G^f
Density	ρ^f
Longitudinal strength allowable	X_L^f
Transverse strength allowable	X_T^f
Out-of-plane shear strength allowable	X_{SOP}^f
In-plane shear strength allowable	X_{SIP}^f
Matrix	
Modulus	E^m
Poisson's ratio	ν^m
Density	ρ^m
Normal strength allowable	X_N^m
Shear strength allowable	X_S^m
Laminate	
Fiber volume fraction	V_f
Total number of plies	n_{layers}



(a) $\theta_1, \theta_2, \theta_3, \theta_4$ (b) V_f (c) $Fiber, Matrix$

Figure 4.—Inputs needed to evaluate a laminate. (a) Laminate structure. (b) Microstructure. (c) Material properties.

To optimize the design of a composite, an objective function is needed that balances the ability of the part to meet the loading conditions and the overall weight of the composite. The objective function used for minimization is defined as

$$\min_x F\,,\ \ F = w_w W_{eff} + w_{TH} THD \tag{11}$$

where W_{eff} is the effective weight of the composite, THD is the Tsai-Hill difference score, and w_w and w_{TH} are weights for the effective weight and Tsai-Hill difference score, respectively. The true weight of the composite cannot be calculated without preselecting the ply geometry which was not done to leave the problem more general (i.e., length and width are unknown, and the thickness of each ply sums to 1 such that $N_x = \sigma_{xx}$). The effective weight instead allows for relative comparisons between candidate configurations, and is calculated as

$$W_{eff} = \left(\rho^f V_f + \rho^m \left(1 - V_f\right)\right) * n_{layers} \tag{12}$$

The Tsai-Hill distance score measures the difference between the evaluated Tsai-Hill score, *TH*, and a defined target Tsai-Hill score, TH_{target}. For most applications, engineers do not design a part to fail at the highest expected load ($TH = 1$), but rather to a predetermined safety factor ($TH_{target} < 1$) to account for errors and discrepancies inherent to material models. The Tsai-Hill distance is therefore calculated as

$$THD = |TH - TH_{target}| \tag{13}$$

This definition of a Tsai-Hill distance allows minimization of the difference between the designed (actual) failure criterion evaluation and the target safety factor without overly penalizing designs that exceed the failure criterion. Designs where TH is slightly greater than TH_{target}, indicating an unsafe structure, are treated more favorably than designs where TH is much less than TH_{target}, indicating an overly safe, heavier design. Thus, Equation (13) looks to drive the design of the structure towards the minimum allowable weight while meeting the defined failure criterion. For this benchmark case, TH_{target} is set equal to 0.9. The weights applied to the effective weight and Tsai-Hill distance score to both account for the difference in scale between the two contributions and change the relative importance of each to the total objective function. For this benchmark problem, the weights are set as w_w=0.01 and w_{TH}= 1.

In most engineering applications, a single load case is not analyzed, but rather a series of different load cases is defined, and the part is designed to meet each. To demonstrate this capability, 4 load cases are defined for the laminate configuration. During optimization, all four load cases produce an objective function score, but only the maximum (i.e., worst) is returned. The four load cases are defined in Table 2.

TABLE 2.—LOAD CASES

Load case	S_{11}, MPa	S_{22} MPa	S_{12} MPa
1 – Pure axial loading	800	0	0
2 – Pure transverse loading	0	200	0
3 – Biaxial loading (2:1 ratio)	600	300	0
4 – Pure shear loading	0	0	500

Magnitudes for each load case were determined such that the failure criteria for the available materials and ply stacking sequences (limited to 8 total plies) would be met in some designs and exceeded in other designs to test the ability to produce optimal, feasible laminate configurations.

Experimental Configuration

For each BO parameter combination (i.e., each chosen combination of benchmark, surrogate, optimization method, acquisition function, and software framework), five tests were run, each using a different random seed for initializing the sample data. Each test was initialized with six randomly sampled input vectors as is the default with Ax, except for the Lolopy tests, which required a minimum of eight initial input vectors. Continuous variables were initialized with Scipy's Sobol sampler, while categorical variables, represented using one-hot encoding, were initialized with Scipy's Latin Hypercube sampler. For hybrid search spaces, continuous and categorical variables were initialized separately using the same random seed and concatenated together into combined input vectors. All experiments were run for a total of 100 samples, including the initial random sampling.

Results

Ishigami Benchmark

The performance of each tested BO parameter combination at each trial number for the Ishigami benchmark is displayed in Figure 4. The mean optimum value, μ_{opt}, and its associated 95% C.I. at trial 100 on the Ishigami benchmark for each BO parameter combination is displayed in Table 3. The same table also shows the mean elapsed running time, μ_t, and its associated 95% C.I. at trial 100 for each surrogate—acquisition function combination. The EI acquisition functions performed the best for this benchmark across most surrogates, on average achieving 95% of the optimum solution value in fewer trials than the other acquisition functions.

Solid Electrolyte Benchmark

The performance of each tested BO parameter combination at each trial number for the solid electrolyte benchmark is displayed in Figure 6. Table 4 shows the performance and computational time (μ_{opt} and μ_t) and their associated 95% C.I. at trial 100 on the electrolyte benchmark for each surrogate—acquisition function combination.

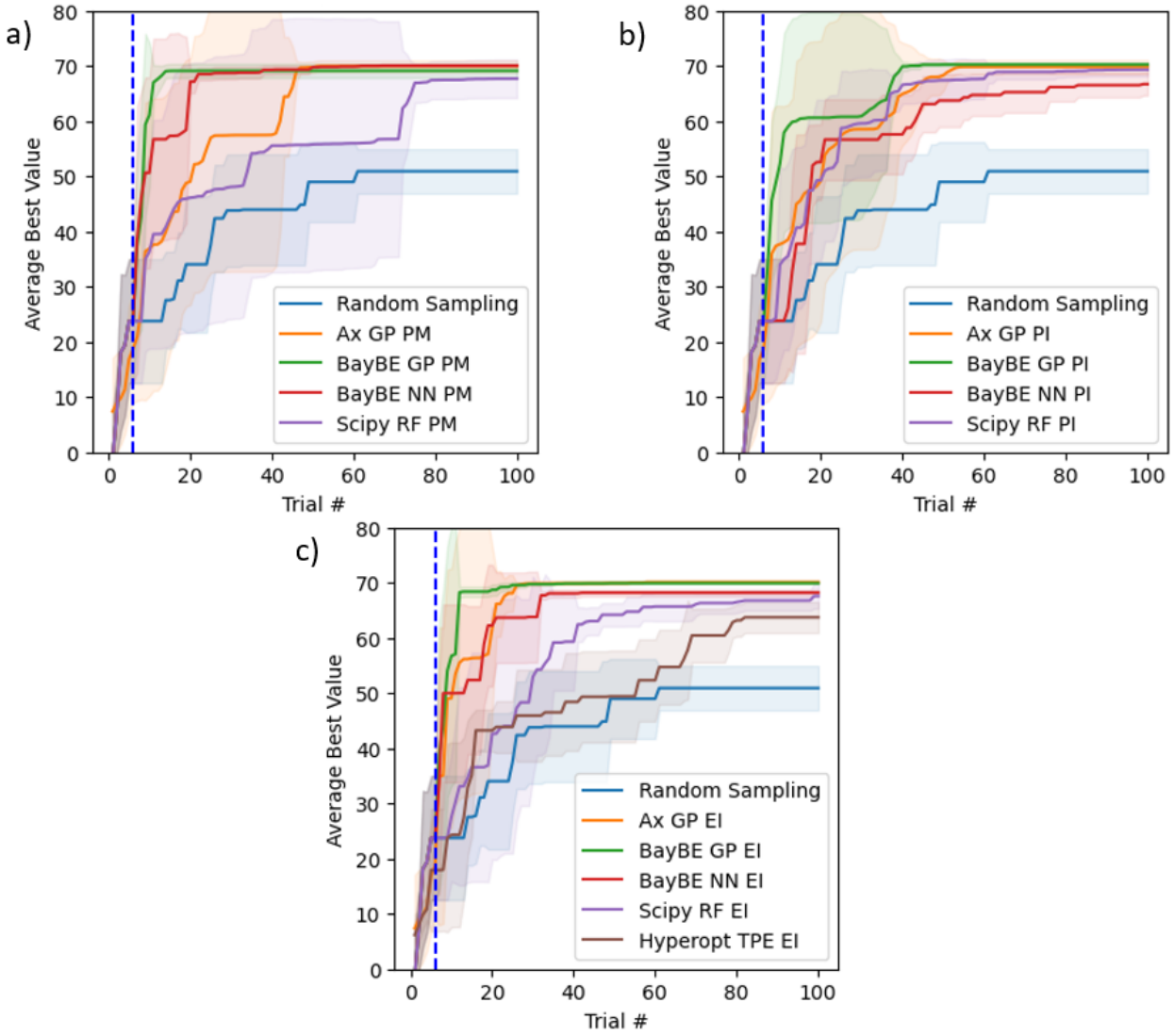


Figure 5.—Plots of Bayesian optimization model performance vs. random sampling on the Ishigami function benchmark using the (a) PM, (b) PI, and (c) EI acquisition function. The vertical blue line indicates the number of initially sampled trials before starting BO. The shaded regions indicate the standard deviation of best values across the 5 trials for each setting.

TABLE 3.—ISHIGAMI BENCHMARK RESULTS, DISPLAYING MEAN OPTIMUM VALUES AND ELAPSED RUNNING TIMES IN SECONDSALONG WITH ASSOCIATED CONFIDENCE INTERVALS

Surrogate/acquisition	μ_{opt} at Trial 100	μ_t at Trial 100, sec
Random	50.921 ± 3.514	0.000558 ± 0.0001
Ax GP - PM	69.975 ± 0.913	56.037 ± 10.831
Ax GP - PI	69.843 ± 1.103	88.374 ± 34.452
Ax GP - EI	70.126 ± 0.206	66.309 ± 4.086
BayBE GP - PM	69.133 ± 1.115	16.248 ± 4.889
BayBE GP - PI	70.266 ± 0.237	34.412 ± 10.325
BayBE GP - EI	69.920 ± 0.0217	20.583 ± 1.327
NN - PM	70.064 ± 0.219	882.01 ± 38.715
NN - PI	66.717 ± 1.796	843.64 ± 292.05
NN - EI	68.194 ± 0.562	826.45 ± 142.77
Lolopy - PM	67.698 ± 3.053	203.61 ± 59.273
Lolopy - PI	69.328 ± 1.026	394.55 ± 148.44
Lolopy - EI	67.599 ± 1.978	305.88 ± 47.161
Hyperopt TPE - EI	63.743 ± 2.462	0.242 ± 0.0082

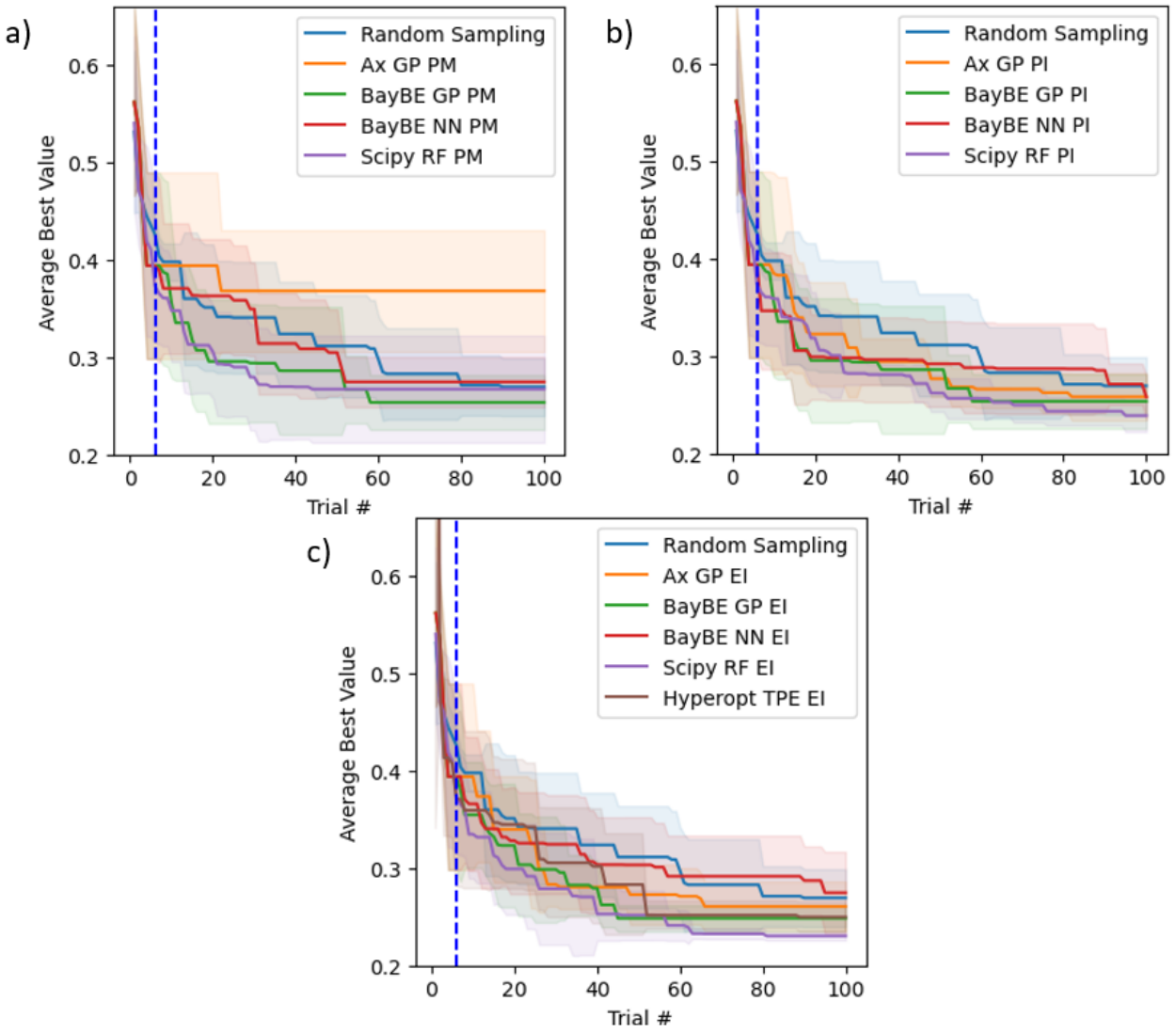


Figure 6.—Plots of Bayesian optimization model performance vs. random sampling on the solid electrolyte benchmark using the (a) PM, (b) PI, and (c) EI acquisition function. The shaded regions indicate the standard deviation of best values across the 5 trials for each setting.

TABLE 4.—SOLID ELECTROLYTE BENCHMARK RESULTS, DISPLAYING MEAN OPTIMUM VALUES AND ELAPSED RUNNING TIMES ALONG WITH ASSOCIATED CONFIDENCE INTERVALS

Surrogate/acquisition	μ_{opt} at Trial 100	μ_t at Trial 100, sec
Random	0.269 ± 0.0259	3.922 ± 0.0436
Ax GP - PM	0.368 ± 0.0546	39.126 ± 1.207
Ax GP - PI	0.258 ± 0.0213	42.543 ± 1.542
Ax GP - EI	0.261 ± 0.0224	43.066 ± 0.308
BayBE GP - PM	0.254 ± 0.0246	21.547 ± 2.313
BayBE GP - PI	0.254 ± 0.0246	25.184 ± 3.262
BayBE GP - EI	0.248 ± 0.0119	25.143 ± 7.852
NN - PM	0.275 ± 0.0226	1023.3 ± 304.19
NN - PI	0.258 ± 0.0199	1016.8 ± 392.17
NN - EI	0.275 ± 0.0369	954.07 ± 302.46
Lolopy - PM	0.267 ± 0.0481	681.58 ± 206.74
Lolopy - PI	0.239 ± 0.0145	565.10 ± 5.280
Lolopy - EI	0.230 ± 0.0039	595.02 ± 9.027
Hyperopt TPE - EI	0.250 ± 0.0133	1.355 ± 0.0212

Composite Material Benchmark

The performance of each tested BO parameter combination at each trial number for the composite benchmark is displayed in Figure 7. The mean optimum value, μ_{opt} and its associated 95% C.I. at trial 100 is displayed in Table 5. The same table also shows the mean elapsed running time, μ_t, and its associated 95% C.I. at trial 100 for each surrogate/acquisition function combination.

Combined Results

Despite the variability experienced by each surrogate—acquisition function combination across different random seeds, some broad conclusions can be drawn.

The neural network surrogate struggled to accurately model the benchmarks with less 100 data points, only outperforming random sampling on the Ishigami function. Neural networks might be expected to provide superior optimization performance only when the dataset size (number of samples and/or feature dimensionality) becomes larger. In addition alternative forms of neural network uncertainty quantification could be tested such as Bayesian neural networks [26], [55], evidentiary heads [56], or neural network ensembles [57].

The PM acquisition function performed the worst compared to PI and EI among all benchmarks. This was likely due to the optimizer getting stuck on local optima, resulting from the completely greedy nature of PM. In contrast, EI frequently yielded near-optimum results with low variability between experiments.

BayBE's GP appeared to act as a generalist, having strong performance across benchmarks and acquisition functions, with the notable exception of PM and PI for the composite benchmark. However, Ax's GP notably underperformed BayBE on benchmarks involving categorical variables. This is due to BayBE's exhaustive sampling of categorical variables across the surrogate, which Ax does not share.

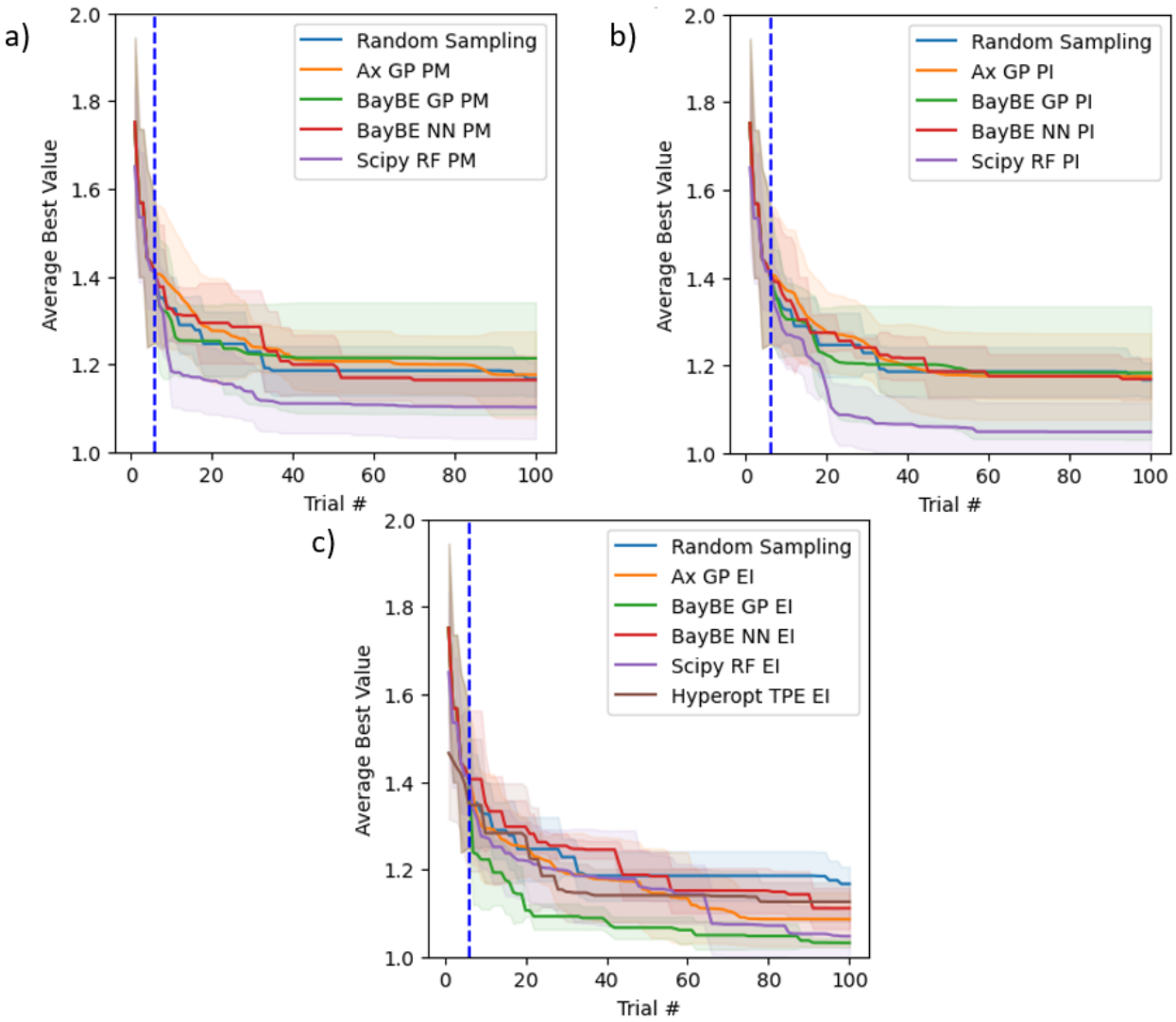


Figure 7.—Plots of Bayesian optimization model performance vs. random sampling on the composite benchmark using the (a) PM, (b) PI, and (c) EI acquisition function. The shaded regions indicate the standard deviation of best values across the 5 trials for each setting.

TABLE 5.—COMPOSITE MATERIAL BENCHMARK RESULTS, DISPLAYING MEAN OPTIMUM VALUES AND ELAPSED RUNNING TIMES ALONG WITH ASSOCIATED CONFIDENCE INTERVALS

Surrogate/Acquisition	μ_{opt} at Trial 100	μ_t at Trial 100, sec
Random	1.167 ± 0.0346	354.33 ± 9.768
Ax GP - PM	1.176 ± 0.0865	563.23 ± 78.514
Ax GP - PI	1.175 ± 0.0859	579.61 ± 113.09
Ax GP - EI	1.086 ± 0.0539	578.41 ± 135.13
BayBE GP - PM	1.213 ± 0.112	1513.9 ± 579.78
BayBE GP - PI	1.183 ± 0.133	1838.9 ± 700.49
BayBE GP - EI	1.032 ± 0.0092	806.74 ± 193.83
NN - PM	1.164 ± 0.0491	3191.3 ± 768.15
NN - PI	1.169 ± 0.0419	3294.1 ± 603.29
NN - EI	1.111 ± 0.0411	2903.4 ± 679.81
Lolopy - PM	1.102 ± 0.0632	7884.9 ± 1288
Lolopy - PI	1.048 ± 0.0589	8350.2 ± 2147.9
Lolopy - EI	1.047 ± 0.0505	7071.4 ± 590.49
Hyperopt TPE - EI	1.126 ± 0.0390	421.42 ± 17.771

Of note was the strong performance of the Lolopy random forest surrogate on benchmarks with categorical variables in their search spaces. The random forests exhibited comparable performance to BayBE GP with significantly less variability on the solid electrolyte benchmark and superior performance to BayBE GP on the composite benchmark, except in combination with the EI acquisition function. This may have been due to a combination of the exhaustive surrogate sampling approach used by the custom optimizer as well as an intrinsic advantage possessed by random forests in processing categorical data. However, the TPE surrogate did not exhibit similar performance on categorical search spaces, perhaps due to Hyperopt's lack of exhaustive surrogate sampling for categorical variables. A summary of the optimization performance of BO parameters (framework, surrogate, and acquisition function) is provided in Table 6.

In terms of execution time, Hyperopt outperformed all surrogates, followed by GPs. BayBE was somewhat slower than Ax on categorical or hybrid search spaces, due to the former's exhaustive sampling of categorical variables over the surrogate. A summary of execution time is provided in Table 7. It should be noted that GPs do not scale well to large numbers of samples (with a computational complexity of $O(N^3)$ for training and $O(N^2)$ for prediction) and would likely exhibit reduced computational efficiency with larger datasets. The random forest and neural network surrogates were the least time performant, with the random forest being especially slow on the composite benchmark, due to a combination of exhaustive categorical variable sampling and the use of the Scipy optimizer. More efficient optimization strategies would likely increase the next candidate selection time for the random forest surrogate. However, in an optimization problem involving physical experimentation, the execution time of the optimization program may be miniscule compared to the total experimental time. Depending on the application, method selection may be driven more by optimization performance than computational complexity.

TABLE 6.—SUMMARY OF OPTIMIZATION PERFORMANCE
Symbols from worst to best are "--", "-", "~", "+", "++".

Framework + Surrogate	Acquisition function		
	PM	PI	EI
Ax GP	-	~	+
BayBE GP	~	+	++
BayBE NN	~	~	~
Scipy RF	~	++	++
Hyperopt TPE	N/A	N/A	-

TABLE 7.—SUMMARY OF EXECUTION TIME
Symbols from slowest to fastest are "--", "-", "~", "+", "++".

Framework + Surrogate	Acquisition function		
	PM	PI	EI
Ax GP	+	+	+
BayBE GP	~	~	+
BayBE NN	--	--	--
Scipy RF	--	--	--
Hyperopt TPE	N/A	N/A	++

Discussion

In this work we have compared BO framework, surrogate, and acquisition function choices across three distinct optimization problems, including two materials related simulations. However, BO in materials science spans a wider range of scenarios than those tacked here. Experiments may contain sparce data with only tens of samples to computational datasets with thousands or millions of samples. The dimensionality of the feature space may span from just a couple features to dozens or orders more. Depending on the experimental execution time, experiments may require next candidate selection in seconds or may prefer to wait much longer if a better candidate selection can be provided. In this section we review reported performance of BO parameter choices across the broader materials science domain.

Performance Under Different Data Conditions

The choice of surrogate can significantly affect performance depending on the number of samples in the dataset. GP models have been a popular default, especially in low-data regimes, because of their strong prior assumptions and data efficiency [1]. In domains with only tens or hundreds of samples (common when each experiment is costly), a GP with an appropriate kernel often achieves excellent predictive accuracy without extensive tuning. However, GPs become computationally expensive as data grow. with Training time scales cubically, $O(n^3)$, with the number of samples, and in practice GPs struggle to handle much more than a few thousand points because of the required covariance matrix inversion. In higher-dimensional feature spaces or with larger datasets, alternative surrogates such as RFs outperform GPs, especially in partially discrete search spaces [58]. The training complexity for tree ensembles grows roughly linearly ($O(n \cdot \log n)$) with data size. Similarly, when ample data are available, neural-network surrogates (including BNNs) become viable or even advantageous. Because of their

ability to learn rich representations, deep neural network surrogates can leverage larger datasets to capture complex trends that a GP might miss [59].

Nonlinearity and Categorical Variables

The nature of the materials optimization problem also influences which surrogate performs best. Standard GPs assume a stationary, smooth function prior, which can underfit sharp or nonstationary phenomena. For example, if the true response has abrupt changes (e.g., phase transitions or threshold effects), a GP with a smooth kernel will tend to "smooth out" those jumps to satisfy its Gaussian assumptions [60]. In contrast, a non-parametric model like a RF has no global smoothness assumption and can more readily capture sudden local variations. In practice, this means that phenomena like piecewise-constant regions or steep cliffs in a processing-property landscape may be represented more faithfully by tree-based surrogates, whereas a GP might erroneously interpolate through them. NN surrogates provide even more flexibility for complex behaviors. They can learn different patterns in different parts of the data and handle multiple outputs in one model.

When categorical or discrete variables are present (for example, categorical choices of processing method or material type in the input), tree-based models tend to have an edge. RF decision trees natively branch on categorical features, whereas a GP would require a specialized kernel or encoding scheme for discrete inputs. Methods like TPE were designed to handle mixed continuous/discrete search spaces by modeling distributions of good versus bad outcomes for each parameter rather than assuming a single smooth response surface. Both RFs and TPE have demonstrated robust performance in high-dimensional or combinatorial optimization settings where standard GPs struggle [58], [59]. Thus, for materials problems involving categorical choices or strongly non-linear behavior, RF, TPE, or neural nets with appropriate architectures often outperform a vanilla GP unless the GP is augmented with a more expressive kernel.

Runtime and Memory Trade-offs

Practical considerations such as runtime and memory usage are also important when selecting a surrogate, especially for automated or real-time experimental loops. GPs impose a significant computational burden as the dataset grows with training time scaling as $O(n^3)$ and memory requirements scaling as $O(n^2)$ [35]. In contrast, tree-based surrogates and neural networks have more favorable scaling. The training time complexity for an RF surrogate is roughly O(n log n) per tree, and neural network training typically scales linearly with number of samples (stochastic gradient descent on mini-batches). With extremely large datasets on the order of 10^5 samples or larger, the GP approach likely becomes intractable while an RF or NN could still be trained in a reasonable time. In scenarios where experimental or simulation throughput is very high (such that the algorithm must propose new conditions rapidly), these differences become significant. A surrogate with faster prediction/training (like RF or TPE) can keep up with a high-frequency loop, whereas a GP might become a bottleneck. However, in many materials optimization cases the evaluation (experiment or computation) is the rate-limiting step, taking minutes or hours, while model fitting takes seconds, in which case the extra overhead of a GP can be acceptable. Thus, the trade-off is context-dependent: if computation time for the surrogate is negligible relative to experiment time, one can afford a more computationally complex model, but if decisions need to be made on the fly or thousands of evaluations are possible, the scalability of the surrogate becomes critical.

Hyperparameter Tuning and Model Assumptions

Each surrogate model comes with its own hyperparameters and implicit assumptions, which affect ease of use and performance.

GPs have relatively few hyperparameters (typically one chooses a kernel and lets the GP optimize its length scale and variance), but they do make strong assumptions through the kernel choice. Selecting an inappropriate kernel (for example, one that enforces excessive smoothness or isotropy) can hinder optimization by misrepresenting the true landscape [61]. In materials problems, researchers often default to kernels like Matérn or radial basis function (RBF), which assume a certain level of smoothness/ differentiability in the objective [60]. If the actual property landscape is rough or has varying length scales in different dimensions, a fixed kernel can slow down convergence until it adapts.

Contrastingly, RFs make very few assumptions about functional form which is one reason they perform well across many domains without much tweaking. RFs do have hyperparameters (number of trees, max depth, etc.), but these are not very sensitive in practice. As long as a reasonable number of trees are used (e.g., 100+), the model tends to be stable and one doesn't usually need to carefully tune hyperparameters to achieve decent performance. This means an RF can be deployed "off the shelf" with default settings and still yield good results, an attractive property when setting up a new autonomous experiment. TPE surrogates are similarly user-friendly with their main hyperparameter being the quantile of observations considered "good," which is often kept at a default (e.g., 10% to 20%).

Neural network surrogates and BNNs involve more tuning (e.g., network architecture, learning rate, training epochs). They can demand expert intervention to get right, although recent progress in automated machine learning and robust Bayesian neural networks is reducing this burden [59].

In general, GPs offer a clean probabilistic foundation with intrinsic uncertainty quantification, while RF and NN surrogates rely on bagging or approximate Bayesian methods to quantify uncertainty. This difference means GPs often give more straightforward uncertainty estimates, while with other models, care must be taken to get comparable confidence measures. Depending on the application's tolerance for uncertainty calibration error, this could be another factor in surrogate selection. Overall, from a practical standpoint, RFs and TPE tend to require less tuning and impose fewer a priori assumptions about the material system, while GPs can serve as a good default choice if a good initial kernel is chosen, and NNs can model more complex relationships at the cost of requiring more hyperparameter tuning.

Benchmark Studies and Practical Recommendations

Recent benchmarking studies provide concrete guidance on when different surrogate models excel. Liang et al. conducted a head-to-head comparison of GP and RF surrogates (among others) on five experimental materials optimization problems [60]. They found that a GP with automatic relevance detection (GP-ARD) and a random forest yielded comparable BO performance in all cases, and both significantly outperformed a baseline GP with an isotropic kernel. The GP-ARD model was marginally more robust, but the RF was a close second and required far less a priori assumption and hyperparameter tuning. This study showed that commonly used GPs can be rivaled by simpler, distribution-free models like RF, especially if the GP's kernel is not well-tailored to the problem.

Li et al. (2023) performed an extensive evaluation of NN surrogates against traditional GPs [59]. They found that standard GPs with Matérn or RBF kernels were surprisingly competitive on many benchmarks, likely due to their strong built-in priors and exact inference, which pay off in small-data settings. BNN-based models showed advantages in certain challenging scenarios and excelled on high-dimensional tasks. They also observed that UQ through NN ensembles did not perform as well with BO.

This suggests that more principled uncertainty estimation (through GPs, BNNs, or TPE's non-parametric approach) is important.

These studies showed that for relatively low-dimensional problems with limited data, a GP surrogate with an expressive kernel is a safe and often optimal choice. If the design space includes categorical parameters, dozens of variables, or irregular response surfaces, then an alternative surrogate should be considered. RFs are strong candidates for such cases, require minimal tuning, handling mixed data types, and higher dimensions well. They also run faster, which can enable tighter human-in-the-loop or robot-in-the-loop experiment cycles. For problems with a very complex response surface, NN surrogates perform well.

The growing body of benchmark data and open-source frameworks is making it easier to test multiple surrogates, and going forward we can expect adaptive strategies (e.g., switching models on the fly) to further improve optimization outcomes in autonomous materials discovery.

Conclusion

In this work, five surrogate model/framework combinations (an Ax Gaussian process, BayBE Gaussian process, BayBE neural network, Scipy + Lolopy random forest, and Hyperopt TPE) were evaluated on three different benchmarks using three different acquisition functions (posterior mean, probability of improvement, and expected improvement). Probability of improvement and expected improvement functions consistently outperformed the greedy posterior-mean, indicating the need to balance exploration against exploitation. GP surrogates gave the best accuracy-per-evaluation on smooth, low-dimensional or mixed domains, while random forests had slightly superior performance when the variables were predominantly categorical. Neural networks were competitive only on the continuous Ishigami surface and required careful hyperparameter tuning to avoid over-fitting. Neural networks may become more competitive in optimization problems with more samples, higher-dimensional feature spaces, or more complex processing-structure-property response surfaces, particularly when applied with alternative uncertainty quantification methods such as Bayesian neural networks.

BayBE GP in combination with EI was found to be the optimum or near-optimum choice across all benchmarks, with Lolopy RF + EI being a strong option for benchmarks with categorical variables in the search space.

By releasing our benchmark code and datasets, we invite the community to build on these baselines and to explore active learning strategies that incorporate advanced surrogate models and adaptive acquisition functions.

References

[1] Y. Wu, A. Walsh, and A. M. Ganose, "Race to the bottom: Bayesian optimisation for chemical problems," *Digit. Discov.*, vol. 3, no. 6, pp. 1086–1100, Jun. 2024, doi: 10.1039/D3DD00234A.

[2] M. Balandat et al., "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 21524–21538, 2020.

[3] E. Bakshy et al., "AE: A domain-agnostic platform for adaptive experimentation," in *Conference on neural information processing systems*, 2018, pp. 1–8.

[4] J. Bergstra, D. Yamins, and D. Cox, "Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures," vol. 28. PMLR, pp. 115–123, Feb. 13, 2013.

[5] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 2623–2631, Jul. 2019, doi: 10.1145/3292500.3330701.

[6] M. Lindauer et al., "SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter

Optimization," *J. Mach. Learn. Res.*, vol. 23, no. 54, pp. 1–9, 2022.
[7] J. Ling, M. Hutchinson, E. Antono, S. Paradiso, and B. Meredig, "High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates," *Integr. Mater. Manuf. Innov.*, vol. 6, no. 3, pp. 207–217, 2017.
[8] K. Kandasamy et al., "Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly.," *J. Mach. Learn. Res.*, vol. 21, no. 81, pp. 1–27, 2020.
[9] T. Ueno, T. D. Rhone, Z. Hou, T. Mizoguchi, and K. Tsuda, "COMBO: An efficient Bayesian optimization library for materials science," *Mater. Discov.*, vol. 4, pp. 18–21, 2016.
[10] K. C. Felton, J. G. Rittig, and A. A. Lapkin, "Summit: Benchmarking Machine Learning Methods for Reaction Optimisation," *Chemistry-Methods*, vol. 1, no. 2, pp. 116–122, Feb. 2021, doi: 10.1002/CMTD.202000051.
[11] F. Häse et al., "Olympus: a benchmarking framework for noisy optimization and experiment planning," *Mach. Learn. Sci. Technol.*, vol. 2, no. 3, p. 035021, Jul. 2021, doi: 10.1088/2632-2153/ABEDC8.
[12] R. J. Hickman et al., "Atlas: a brain for self-driving laboratories," *Digit. Discov.*, vol. 4, no. 4, pp. 1006–1029, Apr. 2025, doi: 10.1039/D4DD00115J.
[13] B. J. Shields et al., "Bayesian reaction optimization as a tool for chemical synthesis," *Nature*, vol. 590, no. 7844, pp. 89–96, Feb. 2021, doi: 10.1038/S41586-021-03213-Y,.
[14] R.-R. Griffiths et al., "GAUCHE: A Library for Gaussian Processes in Chemistry," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 76923–76946, Dec. 2023.
[15] S. B. Harris, R. Vasudevan, and Y. Liu, "Active oversight and quality control in standard Bayesian optimization for autonomous experiments," *npj Comput. Mater.*, vol. 11, no. 1, pp. 1–9, Dec. 2025, doi: 10.1038/S41524-024-01485-2.
[16] Y. Xian et al., "Unlocking the black box beyond Bayesian global optimization for materials design using reinforcement learning," *npj Comput. Mater.*, vol. 11, no. 1, pp. 1–11, Dec. 2025, doi: 10.1038/S41524-025-01639-W.
[17] Y. Iwasaki, D. Ogawa, M. Kotsugi, and Y. K. Takahashi, "Autonomous materials search using machine learning and ab initio calculations for L10-FePt-based quaternary alloys," *Sci. Technol. Adv. Mater. Methods*, vol. 5, no. 1, Dec. 2025, doi: 10.1080/27660400.2025.2470114.
[18] V. Sabanza-Gil et al., "Best Practices for Multi-Fidelity Bayesian Optimization in Materials and Molecular Research," Oct. 2024.
[19] M. A. McDonald, B. A. Koscher, R. B. Canty, J. Zhang, A. Ning, and K. F. Jensen, "Bayesian Optimization over Multiple Experimental Fidelities Accelerates Automated Discovery of Drug Molecules," *ACS Cent. Sci.*, Feb. 2025, doi: 10.1021/ACSCENTSCI.4C01991.
[20] C. Fare, P. Fenner, M. Benatan, A. Varsi, and E. O. Pyzer-Knapp, "A multi-fidelity machine learning approach to high throughput materials screening," *npj Comput. Mater.*, vol. 8, no. 1, pp. 1–9, Dec. 2022, doi: 10.1038/S41524-022-00947-9;SUBJMETA=1034,298,301,563,638,639;KWRD=MATERIALS+CHEMISTRY,THEORETICAL+CHEMISTRY,THEORY+AND+COMPUTATION.
[21] J. I. Myung, J. R. Deneault, J. Chang, I. Kang, B. Maruyama, and M. A. Pitt, "Multi-objective Bayesian optimization: a case study in material extrusion," *Digit. Discov.*, vol. 4, no. 2, pp. 464–476, Feb. 2025, doi: 10.1039/D4DD00281D.
[22] A. Sundar, X. Tan, S. Hu, and M. C. Gao, "CALPHAD-based Bayesian optimization to accelerate alloy discovery for high-temperature applications," *J. Mater. Res.*, vol. 40, no. 1, pp. 112–122, Jan. 2025, doi: 10.1557/S43578-024-01489-0.
[23] D. Khatamsaz, B. Vela, P. Singh, D. D. Johnson, D. Allaire, and R. Arróyave, "Bayesian optimization with active learning of design constraints using an entropy-based approach," *npj Comput. Mater.*, vol. 9, no. 1, pp. 1–14, Dec. 2023, doi: 10.1038/S41524-023-01006-7.
[24] F. Conrad et al., "Exploring design space: Machine learning for multi-objective materials design optimization with enhanced evaluation strategies," *Comput. Mater. Sci.*, vol. 246, p. 113432, Jan. 2025, doi: 10.1016/J.COMMATSCI.2024.113432.

[25] A. K. Y. Low, E. Vissol-Gaudin, Y.-F. Lim, and K. Hippalgaonkar, “Mapping pareto fronts for efficient multi-objective materials discovery,” *J Mater Inf 2023;311.*, vol. 3, no. 2, p. N/A-N/A, May 2023, doi: 10.20517/JMI.2023.02.
[26] S. I. Allec and Maxim Ziatdinov, “Active and transfer learning with partially Bayesian neural networks for materials and chemicals,” *Digit. Discov.*, vol. 4, no. 5, pp. 1284–1297, May 2025, doi: 10.1039/D5DD00027K.
[27] D. Khatamsaz, R. Neuberger, A. M. Roy, S. H. Zadeh, R. Otis, and R. Arróyave, “A physics informed bayesian optimization approach for material design: application to NiTi shape memory alloys,” *npj Comput. Mater.*, vol. 9, no. 1, pp. 1–11, Dec. 2023, doi: 10.1038/S41524-023-01173-7.
[28] S. M. A. A. Alvi, J. Janssen, D. Khatamsaz, D. Perez, D. Allaire, and R. Arróyave, “Hierarchical Gaussian process-based Bayesian optimization for materials discovery in high entropy alloy spaces,” *Acta Mater.*, vol. 289, Mar. 2025, doi: 10.1016/J.ACTAMAT.2025.120908.
[29] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,” *33rd Int. Conf. Mach. Learn. ICML 2016*, vol. 3, pp. 1651–1660, Jun. 2015.
[30] T. K. Ho, “Random decision forests,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, pp. 278–282, 1995, doi: 10.1109/ICDAR.1995.598994.
[31] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[32] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for Hyper-Parameter Optimization,” *Adv. Neural Inf. Process. Syst.*, vol. 24, 2011.
[33] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.
[34] M. Fitzner et al., “BayBE: a Bayesian Back End for experimental planning in the low-to-no-data regime,” *Digit. Discov.*, 2025, doi: 10.1039/D5DD00050E.
[35] C. E. Rasmussen and C. K. I. Williams, “Gaussian Processes for Machine Learning,” *Gaussian Process. Mach. Learn.*, Nov. 2005, doi: 10.7551/MITPRESS/3206.001.0001.
[36] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, “GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration,” *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
[37] S. Elfwing, E. Uchibe, and K. Doya, “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning,” *Neural Networks*, vol. 107, pp. 3–11, Nov. 2018, doi: 10.1016/j.neunet.2017.12.012.
[38] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 2951–2959, 2012.
[39] L. Wen, X. Ye, and L. Gao, “A new automatic machine learning based hyperparameter optimization for workpiece quality prediction,” *Meas. Control (United Kingdom)*, vol. 53, no. 7–8, pp. 1088–1098, Aug. 2020, doi: 10.1177/0020294020932347.
[40] T. Ishigami and T. Homma, “An importance quantification technique in uncertainty analysis for computer models,” in *Proceedings of ISUMA 1990 - 1st International Symposium on Uncertainty Modeling and Analysis*, 1990, pp. 398–403, doi: 10.1109/ISUMA.1990.151285.
[41] A. Y. S. Eng et al., “Theory-guided experimental design in battery materials research,” *Sci. Adv.*, vol. 8, no. 19, p. 2422, May 2022, doi: 10.1126/SCIADV.ABM2422.
[42] A. Urban, D. H. Seo, and G. Ceder, “Computational understanding of Li-ion batteries,” *npj Comput. Mater. 2016 21*, vol. 2, no. 1, pp. 1–13, Mar. 2016, doi: 10.1038/npjcompumats.2016.2.
[43] D. Wang et al., “New insights into Li-argyrodite solid-state electrolytes based on doping strategies,” *Coord. Chem. Rev.*, vol. 508, p. 215776, Jun. 2024, doi: 10.1016/J.CCR.2024.215776.
[44] Y. J. Choi, S. I. Kim, M. Son, J. W. Lee, and D. H. Lee, “Cl- and Al-Doped Argyrodite Solid Electrolyte Li6PS5Cl for All-Solid-State Lithium Batteries with Improved Ionic Conductivity,” *Nanomaterials*, vol. 12, no. 24, p. 4355, Dec. 2022, doi: 10.3390/NANO12244355/S1.
[45] Z. Sun et al., “Insights on the Properties of the O-Doped Argyrodite Sulfide Solid Electrolytes (Li6PS5- xClOx,x=0-1),” *ACS Appl. Mater. Interfaces*, vol. 13, no. 46, pp. 54924–54935, Nov. 2021, doi: 10.1021/ACSAMI.1C14573.

[46] W. Arnold et al., “Synthesis of Fluorine-Doped Lithium Argyrodite Solid Electrolytes for Solid-State Lithium Metal Batteries,” *ACS Appl. Mater. Interfaces*, vol. 14, no. 9, pp. 11483–11492, Mar. 2022, doi: 10.1021/ACSAMI.1C24468.
[47] L. Zhou, A. Assoud, Q. Zhang, X. Wu, and L. F. Nazar, “New Family of Argyrodite Thioantimonate Lithium Superionic Conductors,” *J. Am. Chem. Soc.*, vol. 141, no. 48, pp. 19002–19013, Dec. 2019, doi: 10.1021/JACS.9B08357.
[48] C. Chen and S. P. Ong, “A universal graph deep learning interatomic potential for the periodic table,” *Nat. Comput. Sci.*, vol. 2, no. 11, pp. 718–728, Nov. 2022, doi: 10.1038/S43588-022-00349-3.
[49] L. L. Wong, K. C. Phuah, R. Dai, H. Chen, W. S. Chew, and S. Adams, “Bond Valence Pathway Analyzer-An Automatic Rapid Screening Tool for Fast Ion Conductors within softBV,” *Chem. Mater.*, vol. 33, no. 2, pp. 625–641, Jan. 2021, doi: 10.1021/ACS.CHEMMATER.0C03893.
[50] S. J. Honrao et al., “Discovery of novel Li SSE and anode coatings using interpretable machine learning and high-throughput multi-property screening,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–14, Dec. 2021, doi: 10.1038/S41598-021-94275-5.
[51] B. A. Bednarcyk and S. M. Arnold, “MAC/GMC 4.0 User’s Manual: Keywords Manual. Volume 2,” 2002.
[52] M. Paley and J. Aboudi, “Micromechanical analysis of composites by the generalized cells model,” *Mech. Mater.*, vol. 14, no. 2, pp. 127–139, 1992.
[53] J. Aboudi, M. J. Pindera, and S. M. Arnold, “Higher-order theory for periodic multiphase materials with inelastic phases,” *Int. J. Plast.*, vol. 19, no. 6, pp. 805–847, Jun. 2003, doi: 10.1016/S0749-6419(02)00007-4.
[54] B. Raju, S. R. Hiremath, and D. Roy Mahapatra, “A review of micromechanics based models for effective elastic properties of reinforced polymer matrix composites,” *Compos. Struct.*, vol. 204, pp. 607–619, Nov. 2018, doi: 10.1016/j.compstruct.2018.07.125.
[55] E. Goan and C. Fookes, “Bayesian Neural Networks: An Introduction and Survey,” *Lect. Notes Math.*, vol. 2259, pp. 45–87, Jun. 2020, doi: 10.1007/978-3-030-42553-1_3.
[56] M. Sensoy, L. Kaplan, and M. Kandemir, “Evidential Deep Learning to Quantify Classification Uncertainty,” *Adv. Neural Inf. Process. Syst.*, vol. 2018-Decem, pp. 3179–3189, Jun. 2018.
[57] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, pp. 6403–6414, Dec. 2016.
[58] J. van Hoof and J. Vanschoren, “Hyperboost: Hyperparameter Optimization by Gradient Boosting surrogate models,” Jan. 2021, Accessed: Jun. 20, 2025. [Online]. Available: https://arxiv.org/pdf/2101.02289.
[59] Y. L. Li, T. G. J. Rudner, and A. G. Wilson, “A Study of Bayesian Neural Network Surrogates for Bayesian Optimization,” *12th Int. Conf. Learn. Represent. ICLR 2024*, May 2023, Accessed: Jun. 13, 2025. [Online]. Available: https://arxiv.org/pdf/2305.20028.
[60] Q. Liang et al., “Benchmarking the performance of Bayesian optimization across multiple experimental materials science domains,” *npj Comput. Mater.*, vol. 7, no. 1, pp. 1–10, Dec. 2021, doi: 10.1038/S41524-021-00656-9;SUBJMETA=166,301,639;KWRD=ENGINEERING,MATERIALS+SCIENCE.
[61] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, “Taking the human out of the loop: A review of Bayesian optimization,” *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016, doi: 10.1109/JPROC.2015.2494218.