Integrating Materials and Manufacturing Innovation (2023) 12:157–170
https://doi.org/10.1007/s40192-023-00301-x

TECHNICAL ARTICLE

Faux-Data Injection Optimization for Accelerating Data-Driven Discovery of Materials

Abdul Wahab Ziaullah[1] · Sanjay Chawla[2] · Fedwa El-Mellouhi[1]

Received: 16 September 2022 / Accepted: 16 May 2023 / Published online: 5 June 2023

Abstract

Artificial intelligence is now extensively being used to optimize and discover novel materials through data-driven search. The search space for the material to be discovered is usually so large, that it renders manual optimization impractical. This is where data-driven search and optimization enables us to resourcefully locate an optimal or acceptable material configuration with desirable target properties. One such prominent data-driven optimization technique is Bayesian optimization (BO). Among the mechanics of a BO is the use of a machine learning (ML) model that learns about the scope of the problem through data being acquired on the fly. In this way a BO becomes more informative, directing the search more exquisitely by providing informative suggestions for locating a suitable material candidate for further evaluation. The candidate material is suggested by proposing parameters such as its composition and configuration, which are then evaluated either by physically synthesizing the material and testing its properties or through computational methods such as through density functional theory (DFT). DFT enables researchers to exploit massively parallel architectures such as high-performance computing (HPC) which a traditional BO might not be able to fully leverage due to their typical sequential data-acquisition bottleneck. Here, we tackle such shortcomings of BO and maximize the utilization of HPC by enabling BO to suggest multiple candidate material suggestions for DFT evaluations at once, which can then be distributed in multiple compute nodes of an HPC. We achieve this objective through a batch optimization technique based on faux-data injection in the BO loop. In the approach at each candidate suggestion from a typical BO loop, we "predict" the outcome, instead of running the actual experiment or DFT calculation, forming a "faux-data-point" and injecting it back to update an ML model. The next BO suggestion is therefore conditioned on the actual data as well as faux-data, to yield the next candidate data-point suggestion. The objective of this methodology is to simulate a time-consuming sequential data-gathering process and approximate the next k-potential candidates, quickly. All these k-potential candidates can then be distributed to run in parallel in an HPC. Our objective in this work is to test the theory if faux-data injection methodology enables us accelerate our data-driven material discovery workflow. To this end, we execute computational experiments by utilizing organic–inorganic halide perovskites as a case study since the optimality of the results can be easily verified from our previous work. To evaluate the performance, we propose a metric that considers and consolidates acceleration along with the quality of the results such as the best value reached in the process. We also utilize a different performance indicator for situations where the desired outcome is not material with optimal properties but rather a material whose properties satisfy some minimum requirements. We use these performance indicators to compare this BO-based faux-data injection method (FDI-BO) with different baselines. The results show that based on our design constraints, the FDI-BO approach enabled us to obtain around two- to sixfold acceleration on average compared to the sequential BO.

Keywords Bayesian optimization · Artificial intelligence · Data-driven machine learning · Materials discovery

Introduction

Recent advancements in artificial intelligence (AI) have paved the way for accelerating the discovery of novel materials [38, 50]. Among these advancements are data-driven approaches [24], which utilize machine learning (ML)

Extended author information available on the last page of the article

models and optimization algorithms to make informed decisions about potential candidate material that might have the desired target properties. The burden is then left with verification methods such as laboratory synthesis/testing or computational methods to determine if the suggested material is indeed the target material being sought. In case the material is verified as not optimal or suitable, its data is assimilated for an improved ML model and the process continues until the desired material is discovered or the resources are exhausted. The use of such data-driven discovery methods has enabled the automation of experiments such as through the use of self-driving labs [2]. A typical data-driven method either attempts to enrich the data with more representative distribution to improve the prediction capabilities of the underlying ML model, with a process called active Learning [52] or attempts to locate global optima, by acquiring as minimal data as possible. Through a data-driven process, the underlying ML model consequently becomes more accurate in predicting properties of materials that have not yet been synthesized or discovered, and some of the material properties these models help predict include band gap [9, 15, 64] formation energies [11, 39], phase stability [16, 25], crystal structures [55], etc. Through this use of AI, researchers are able to eliminate the significant burden of laboratory synthesis, molecular simulations, or DFT calculations. Moreover, DFT calculations scale poorly with the size of the system [23], further motivating the use of AI to accelerate the process.

Although these ML models are orders of magnitude faster compared to traditional DFT calculations, they rely on niche data distribution [12], such as, for a specific use case, electron correlated data [13], to yield out-of-sample prediction accuracy that is on par with the resolution required for decision making. Often for novel materials, such niche distribution of data might not be available; therefore, the accuracy of ML predictions might not be close to the resolution required to suggest the candidate material with desired properties.

All these problems sufficiently motivate a faster data-driven approach to acquire data more representative of the target problem through highly selective candidate material suggestions. Bayesian optimization is a widely used approach for such data-driven materials discovery [14, 44, 46], but it suffers from a sequential execution bottleneck. To accelerate the process, one can exploit batch optimization techniques where several candidate parameters of the material can be selected at once for verification and assimilation through several distributed DFT calculations or laboratory experiments. To this end, we focus on a batch optimization technique based on the faux-data injection method and compare its performance against other baselines.

This work adopts the use case from our previous work on sequential Bayesian optimization on the perovskite family of materials [46]. Halide perovskites offer huge potential for compositional and structural tuning enabling the possibility of discovering high-performing materials for energy conversion. Hybrid halide perovskites (ABX_3) consist of organic molecules sitting at its A-site; a metal *Ge*, *Pb*, or *Sn* at its B-site; and *I*, *Br*, and *Cl* at its X-site [59, 63] as well as mixture of the above constituents. In our work, we utilize methylammonium lead halide ($CH_3NH_3PbI_3$) as the baseline for the perovskite family, due to its suitable band gap and ease of synthesis. A comprehensive list of various ML models on the perovskite family of materials is presented in Tao et al. [57] to predict properties such as band gap, formation energy, formability, and stability. Our analysis, however, is based on DFT calculations to determine the enthalpy of mixing [44], to predict the phase stability of mixed perovskites indicative of how likely will two stable hybrid perovskites compounds mix to form a homogenous solid solution.

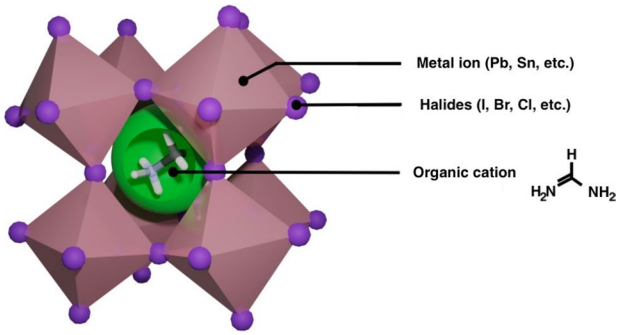


Fig. 1 Composition of perovskite lattice

Background

Materials Discovery-Perovskite Use Case

The design and discovery of materials are often multi-objective criteria and require optimization to yield structures that simultaneously have better production/conversion yield, stability, and other domain-specific properties. Perovskites are used in solar cells as they belong to the family of thin-film solar cells with good prospects to emerge in the market for niche applications [21]. The optimization of the perovskite's light-absorbing layer revolves around finding suitable compositions of metal ions, halides, and cations to enable good power conversion properties while ensuring a long time cell stability (Fig. 1).

This optimization criterion is well suited for our use case as we can restrict to certain known metal ions, halides, and cations and explore mainly the configuration space spanned by these substitutes. In our use case, we restricted the substitutes to methylammonium/ethylammonium lead halide $(CH_3NH_3)_x(C_2H_5NH_3)_{1-x}PbI_3$) as their configuration was

exhaustively explored in our previous paper [46]. Restricting these substitutes simplifies the problem of finding concentrations of methylammonium and ethylammonium and their corresponding orientations that produce a more stable configuration corresponding to the lowest enthalpy of mixing (ΔH_{mix}) that favors forming a homogeneous solid solution that would not easily segregate into the parent single cation methylammonium and ethylammonium compounds.

The optimization for such materials is often carried out by DFT and/or molecular dynamics calculations which are considered a faster and cheaper alternative to laboratory synthesis. However, DFT computations are still comparatively more time-consuming than ML predictions. For single DFT computation of $2 \times 2 \times 2$ supercell size of methylammonium/ethylammonium lead halide on modern HPC running on a single node with 48 CPUs, it takes anywhere from 35 to 70 min to complete one VASP calculation using the PBE functional.

For a $3 \times 3 \times 3$ supercell containing a given mixture of MA and EA cations, there are 27 cationic sites that can be populated by 13 unique combinations per site resulting in ${}^{27}C_{13} = \frac{27!}{13!(27-13)!} = 20{,}058{,}300$ possible configurations [46]. Assuming it takes 1 h of CPU time for running a DFT on a single configuration, the time needed to exhaustively evaluate all the configurations would take a couple of thousand years. Such a cost calls for a more principled approach to sample from the search space. In the next section, we discuss two main approaches that are often leveraged for material discovery.

Human-in-the-Loop Approach

A typical human-in-the-loop approach relies on an expert's knowledge or intuition with optionally some scientific model to guide the exploration and exploitation of potential candidate materials. The control flow of such an approach is presented in Fig. 2 and starts with prior knowledge of the expert based on some scientific model or data, which is then utilized to propose a suitable candidate composition for DFT calculations. The result from the calculation is either statistically or qualitatively analyzed which may prove or disprove the expert's hypothesis. Based on the observation, the expert updates his/her hypothesis either cognitively or through a model for the suggestion of the next candidate material. This approach with human-in-the-loop may be good as far as "cognitive learning" of the domain is concerned; however, as pointed out by Park et al. [46], this could lead to adding an induced bias of the expert, which may restrict him/her from exploring other potentially informative configuration. Limitation of the inherent cognitive bias of human-in-the-loop and its comparison for machine-centered designs of materials is presented in Peng et al. [47], covering

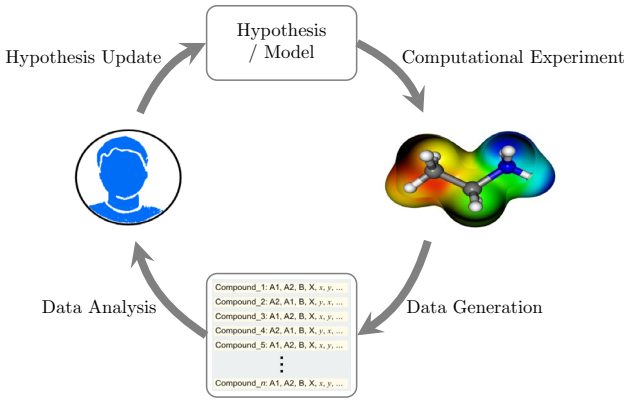


Fig. 2 Human-in-the-loop method to material optimization: an initial hypothesis or model is used to set up an experiment, and the generated data is then analyzed, resulting in knowledge update from an expert, which in turn enables model enhancement to determine the next best experiment

model-based systems, descriptor-based systems, data-driven methods including active learning, Bayesian optimization, and inverse design.

Moreover, the typical human-in-the-loop approach suffers from the requirements for the availability of an expert as well as cognitive delays attributed to obtaining knowledge from the data, performing analysis, and establishing renewed hypotheses for further action.

Data-Driven Approach

The data-driven approach enables a more principled way of searching for optimal or suitable material configuration. They, however, rely on the quality of data-acquisition models. There are several optimization methods that are used to search for optima of some black-box functions. We can classify them into sequential or parallel methods. Bayesian optimization, reinforcement learning, and the Markov-Chain-Monte-Carlo algorithm are a few examples of sequential optimization, while particle swarm optimization, simulated annealing, and ant-colony optimization are examples of batch/parallel optimization. Most data-driven approaches utilize a mix of pseudo-random techniques along with exploration and exploitation strategies to locate global best values. While some methods aim to seek the global optima which are useful for tasks like finding metal–organic framework (MOF) that is highly selective for carbon capture and conversion, other methods such as active learning aim to seek data enrichment to reduce uncertainty in the ML model for tasks such as querying the properties of a given material configuration.

Depending upon the nature of these data-driven methods, either all of the data is utilized to determine the next candidate for an evaluation such as the case with Bayesian

optimization, or the best value in the data acquired so far is utilized to compute the next candidate evaluation such as some variants of PSO or only current data-point is utilized such as Markov-Chain-Monte-Carlo algorithm. Each of these has its advantages and drawbacks and we shall focus on a few in the next section.

On the other hand, data-agnostic methods either randomly acquire new data-point or exhaustively sweep the search space. This often poses a challenge should the cost of running an experiment or DFT computation to acquire new data be high, rendering complete reliance on data-agnostic methods for obtaining global best values, impractical.

Among various data-driven methods that optimize the data-gathering process, we focus on particle swarm optimization, simulated annealing, genetic algorithms, and Bayesian optimization as they are widely used in computational materials science and are suitable for running in parallel nodes of an HPC.

Particle Swarm Optimization (PSO)

PSO is a bio-inspired model-free approach where several particle "swarms" are utilized for optimizing some target objective. PSO algorithm essentially enables several parameters to be evaluated in a batch, where each particle is a single evaluation. In its vanilla algorithm, each particle keeps track of the best value found by that particle as well as the swarm, to determine its next location for parameter evaluation. Collectively, the swarm can find the global best value. There are also other variants of PSO that make use of more information than just the best value [20, 36].

PSO has been utilized in the discovery and optimization of materials such as the discovery of new polymorphs of gallium oxides [61], here PSO was used in conjunction with the first-principles crystal structure prediction method to find the most stable structures.

A comparative study [56] between Bayesian optimization and PSO shows that BO outperforms PSO based on a minimally acquiring the informative data to converge to optima, which is desirable for real-world problems as there is usually a cost associated with data acquisition. This motivates the utilization of Bayesian optimization as the method of choice in various use cases.

Simulated Annealing (SA)

SA is inspired by the thermodynamic process of metal annealing and is suitable to find global optima in a multimodel function. SA uses probabilistic jumps which is a function of theoretical temperature, and as the temperature is lowered: "process of annealing," the probability of jumps decreases in the search space [27].

Adoption of SA algorithm in the materials science domain ranges from computational exploration of potential energy landscapes [29, 43] to the design optimization of thin-film systems [5] where SA was used to optimize various parameters of the thin-film system such as desired reflectance, thickness, and the refractive index of each layer, with k-total layers of the system. In a recent study, SA has been utilized to optimize global energy in zeolite framework systems [1] where it was utilized along with DFT simulations to locate global minima energy by ramping up and down the temperature parameters of SA. SA has also been extended to peptides where the lowest conformation energy of various peptides was found [62]. Further review and extension of SA to other domains are presented by Kirkpatrick et al. [30].

Genetic Algorithms (GA)

Genetic algorithms are population-based algorithms where a criterion for good features is set using a fitness function and is inherited in the next generation. Genetic algorithms provide a way to carry forward those parameters that are evaluated to achieve better results and in the next generation, they are cross-bred and mutated so that they can be further explored.

Genetic algorithms have been adapted for both data-driven material discovery and properties prediction. Researchers have utilized GA to search composite material by converting the material descriptors into an array of genes and utilizing GA to obtain the composite that has desirable properties, they also utilize GA for predicting the mechanical properties of composite materials [40]. Ikeda et al. utilized GA in molecular dynamics simulations to discover the optimal composition of an alloy Ni-Al-Cr-Mo-Ta and Ni-Al-Cr-Mo-Ta based on some design rules [26].

GA can also be used with other optimization method and have been used in conjunction with PSO to discover novel phosphorous for yellow-LEDs [54]. GA was utilized for preliminary screening in a 10-dimensional composition search space followed by refinement in a 6-dimensional search space using PSO. An exhaustive survey of GA applied to materials optimization is provided in Charkaborti et al. [4].

Bayesian Optimization (BO)

Bayesian optimization (BO) addresses the objective of finding global optima of the target problem by sequentially suggesting candidate parameters to be evaluated [53]. The target problem is approximated by using some surrogate model. BO uses ML models such as Gaussian process (GP) or random forest (RF) [37] for the surrogate because they provide uncertainty quantification of variables that BO requires. BO is typically more resourceful as it optimizes the search by forming posterior probability distribution based on the

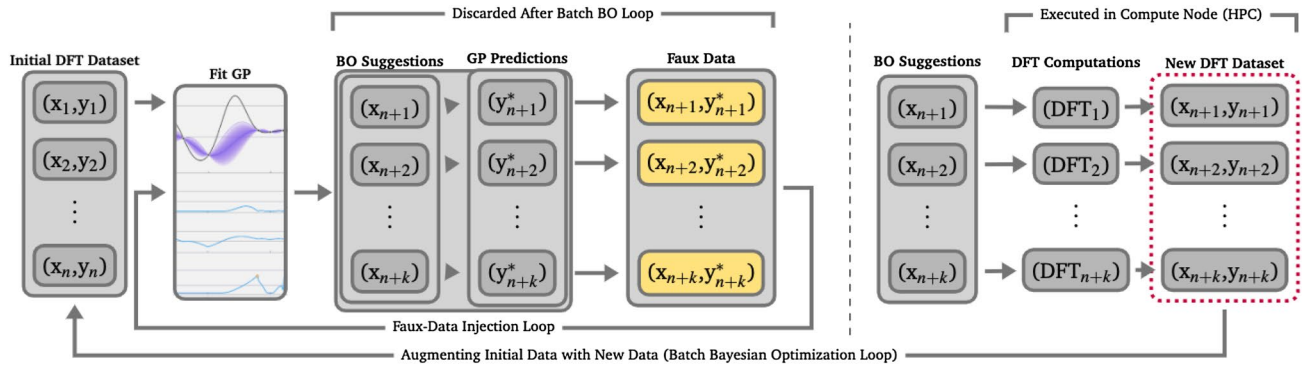


Fig. 3 Flow diagram of faux-data injection method depicting inner faux-data injection loop and outer batch Bayesian optimization loop

updated surrogate model. BO has been adapted to several domains such as environmental mapping [7], civil engineering [51], and materials discovery [38].

A typical BO loop executes as follows: An initial dataset is used to establish a surrogate model. A model would then have prior information on data distribution, current minima/maxima, uncertainties, correlations, etc. BO uses the acquisition function on this model to determine the next candidate parameter, such that the data-point obtained by evaluating this candidate parameter/s is expected to locate some optima of the model or reduce the uncertainty in the model, depending upon the exploration and exploitation strategy of the acquisition function. After a data-point is obtained, it is assimilated in the surrogate model, which in turn enables BO to suggest the next candidate parameter/s to be evaluated. This process is repeated until desired convergence is achieved or the experimental budget is exhausted.

Here, parameter/s evaluation implies determining target property such as the band gap of a material given the concentration of its constituent molecules (parameters), yielding a data-point. We could either use DFT to perform such evaluation or use laboratory synthesis and testing. BO is only concerned with obtaining the data, irrespective of the evaluation method.

Assuming the availability of multiple computation resources within an HPC environment, a typical BO loop may not be able to leverage its potential due to its sequential execution bottleneck. To parallelize the process, we utilize faux-data at each of the parameter evaluation steps of the BO, to obtain the next suggestion and consequently build a batch of candidate parameters to be evaluated.

Parallelizing BO has been done using various methodologies [6, 19, 22, 60] In this paper, we consider faux-data injection in BO loop which is based on Kriging believer heuristic [17]. However, unlike Kriging believer we also utilize a random suggestion using an ϵ-greedy algorithm. This also helps to unbias the suggestions, should the GP model be less informative.

The success of BO based on its extensive adoption of computational materials science is attributed to the fact that while other methods such as PSO, GA & SA utilize pseudo-random methods to converge to desired optima, BO is more methodical and resourceful. In computational materials science methods such as molecular dynamics simulations and DFT, the evaluation of a data-point comes with a high computational cost, rendering random and pseudo-random methods less attractive. We, therefore, adapt it as a starting point for our study.

Faux-Data Injection in Bayesian Optimization

Faux-data injection methodology, elaborated in Fig. 3, involves injecting faux-data into the Bayesian optimization loop. As mentioned in the previous section, in a typical BO loop, a candidate parameter evaluation is suggested given some prior surrogate model. The model is augmented once a new data-point is added to the existing dataset. The augmented model is utilized again to suggest a new candidate location, the process is then repeated until the budget is exhausted or some error criterion is satisfied. In the faux-data injection approach, we bypass the parameter evaluation process after the candidate location is suggested by BO; and utilize a "predicted value" at that suggested location, rather than running actual DFT or experimental evaluation. This is represented by the "Faux-Data Injection Loop" in Fig. 3. The predictor, in our case, a Gaussian process, provides the predicted value ⟶ "faux-data-point." This faux-data-point is combined with initial data, to "mimic" an updated dataset and is fed back into the BO loop, to suggest another candidate parameter for evaluation. The process is repeated until a batch of "k" candidate suggestions is achieved. The size of the batch is determined by various criteria and is presented in Sect. 4.5.

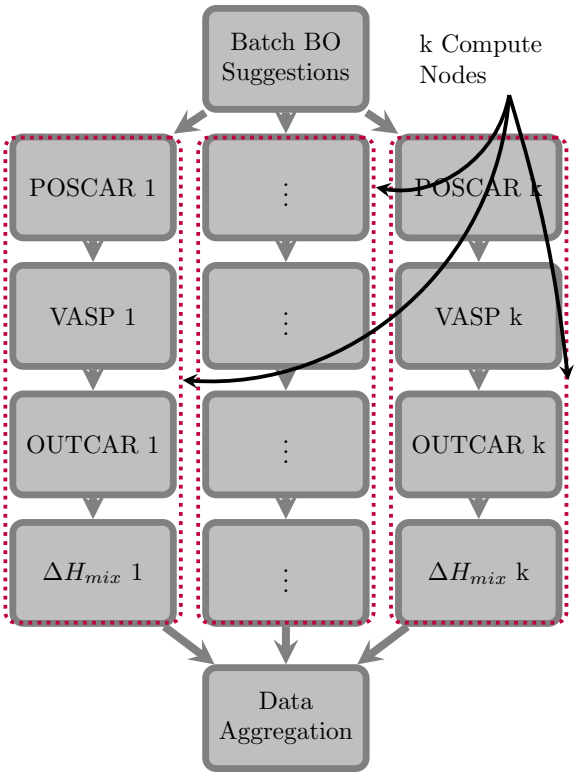


Fig. 4 Close-up on parallel computation in k-compute nodes of HPC

Once a batch of "k" candidate suggestions is obtained, they are sent to the supercomputer to perform actual parameter evaluation as shown in Fig. 4 providing a close-up of "Batch Bayesian Optimization Loop" in Fig. 3. The faux-data that was generated is discarded as soon as the real evaluations are obtained from the DFT calculations. The faux-data injection loop is repeated to get the next "k" candidate suggestions. In our case, we repeat the process until the sampling budget is exhausted as discussed in Sect. 4.4.

The benefit of utilizing the predictor rather than running actual experimentation is to obtain quick-and-shallow initial suggestions. This suggestions generation process is relatively fast taking a few seconds to achieve a batch of five suggestions for evaluation. Another approach to achieving a batch of suggestions is based on local penalization of acquisition function [19]; in this approach, only the acquisition function is penalized where the optima of the acquisition function is flattened so that it is no longer the optima after the suggestion is taken, this enables the optimizer to search for next best optima, which in turn gets penalized after the parameter suggestion is taken. However, the drawback to this approach is that the acquisition function should be multimodal; otherwise, the samples tend to be clustered around certain regions of the exploration space. Another approach involves k-means clustering [22].

In summary, rather than running single DFT computation at a time, we run k-DFT computations to maximize the utilization of computational resources within a high-performance computing environment. The method is adapted from Kriging believer's (KB) heuristic [18] with an additional ϵ-greedy algorithm to unbias the search direction with a frequency of $\epsilon = 0.3$. We evaluate this approach in Sects. 5.1 and 5.2 to demonstrate the performance in the discovery of stable mixed-cation halide perovskites.

HPC Computations, Evaluation, and Benchmarking

We ran the computations in a high-performance computing system. We utilized the Rocketsled library [10] to handle the proposed workflow. One of the features of the Rocketsled library is that it uses MongoDB to handle the data generated in the workflow. This created some limitations, as the compute nodes in all our HPCs were firewalled. MongoDB was only accessible from the login node. Therefore, the workflow was divided between the login node and compute node where a less computational intensive process, "Faux-Data Injection Loop" in Fig. 3, was executed in the login node, and all the expensive DFT computations were offloaded to compute nodes. Rocketsled library provides a multi-threading option; therefore, all the DFT computations were offloaded simultaneously to the compute nodes. Each DFT computation or a "job" was assigned 2000 MB of memory per CPU running 40 tasks per node.

An exhaustive benchmarking of BO applied to material science is presented in reference [37], which serves as the basis for the hyperparameters used in this study, namely Gaussian process for surrogate modeling and expected improvement for acquisition function. To ensure a fair comparison of all the methods in the study, we train the Gaussian process with 10 initial DFT data-points, which are selected randomly to provide the same starting condition to FDI-BO, TOPK-BO, and S-BO.

Our optimization problem is based on DFT calculation of the enthalpy of mixing and the analysis of perovskite structural features to narrow down the compositional search domain for cation mixtures toward concentrations that preserve the perovskite structure while pointing toward the maximal stability

Our group recently demonstrated that the enthalpy of mixing plays a significant role in enhancing the intrinsic stability of mixed-cation halide perovskites compared to the conventional single cation $MAPbI_3$ material [45]. The enthalpy of mixing $APbI_3$ and $A'PbI_3$ (ΔH_{mix}) is calculated according to:

$$\begin{aligned}\Delta H_{\text{mix}}\left(A_{1-x}A'_{x}\text{PbI}_3\right) &= \Delta H\left(A_{1-x}A'_{x}\text{PbI}_3\right) - (1-x)\Delta H\left(A_{1-x}\text{PbI}_3\right) \\ &+ x\Delta H\left(A'_{x}\text{PbI}_3\right)\end{aligned} \quad (1)$$

where ΔH is the enthalpy of formation for a given chemical formula of perovskite obtained from DFT calculations.

Our search of the perovskite structures at which $\Delta H_{\text{mix}} < 0$ meV/ion implies the relaxation of the pure perovskites by the cation mixing. Subsequently, the negative value of ΔH_{mix} suggests a suppressed EA cation segregation within the perovskite structures at the corresponding concentrations, thereby indicating the stability of the material.

DFT Calculation Details

We performed the DFT calculations of the $(2 \times 2 \times 2)$-supercell perovskites, using the Vienna Ab-initio Simulations Package (VASP) [32–34]. The projector-augmented wave [3, 35] method described the wavefunction of the valence electrons at the Perdew–Burke–Ernzerhof (PBE) [48, 49] functional level. The plane-wave cutoff was set at 520 eV for all the calculations. Besides, to make up for the GGA's inadequate dispersion interactions, we include Tkatchenko–Scheffler van der Waals correction [58]. We optimized the structures until the energies and forces converged within 10^{-7} eV/atom and 0.01 eV/Å, respectively. Moreover, the calculated energies of the structures are obtained with Monkhorst–Pack scheme [41] $4 \times 4 \times 4$ k-point grid.

Besides, the orientation of each cation was rotated differently in the initial coordinates. To manage the workflow, we utilized Rocketsled and Fireworks library [10, 28] and Pymatgen [42] tools to set the perovskites and analyze the optimized coordinates. Calculations were carried out on a Cray XC50 HPC system with Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz running Linux. The system consists of 4800 Intel Xeon Skylake Gold CPUs with 25.344 TB aggregated memory. The workload was distributed as follows: 1-DFT computation per node with 40 CPUs and 1 thread dedicated to OpenMP support.

Evaluation Criteria

We designed the computational experiments as a proof-of-concept to show the acceleration that can be achieved by the faux-data injection method through the use case of determining the optimal concentration of mixed organic–cation halide perovskite.

There are various baseline methodologies and performance indicator metrics to compare the performance of FDI-BO with; however, due to limited resources we selected those that are more relevant and have widely been implemented in other relevant literature, such as sequential BO [53]. We also utilized, TOPK-BO [10], where at each BO loop, a batch of TOPK-best suggestions are naively selected and as a control, we also used random sampling. Although various studies [8, 37] use random sampling as a reference for comparison, in our case as we are interested to determine the performance of faux-data-based parallelization of our workflow, we utilized S-BO as the reference baseline which was used in our previous work. For the choice of performance indicator, we utilized two metrics, namely acceleration factor [37] (AF) and our relative overall performance (ROP) metric. AF quantifies acceleration with respect to some reference method via the time frame required to reach a certain threshold value, rather than optimal or best value (2). It is a single objective metric, which disregards the best value or optimal value, in favor of some threshold value. This might be useful in cases where a material optimization below some threshold value is acceptable, such as a material with some minimum bulk modulus, and minimum temperature resistance. However, in many other applications, we are interested in pushing the limits of the materials by searching for configurations with optimal or best values. Therefore, to address such requirements we developed ROP metric that consolidates best values along with the acceleration, and is discussed in the next section.

In our study, as we are comparing parallel methods (FDI-BO, TOPK-BO) against the reference sequential BO, we utilize batch cycles to calculate the acceleration. In a single batch cycle, FDI-BO and TOPK-BO would execute k-DFT evaluations, while S-BO and random sampling will execute only 1-DFT evaluation. We modified the original AF formula to reflect this in Eq. (2):

$$\text{AF} = \frac{\text{batch cycle to reach cut-off by (a)}}{\text{batch cycle to reach cut-off by (b)}} \quad (2)$$

Here, (a) = FDI-BO, TOPK-BO, RS as a target, while, (b) = S-BO as a reference. AF enables us to determine how much faster one could reach a specific threshold value compared to reaching the same threshold by the reference approach. For the sake of analysis, we used thresholds: -1, -2, and -3 meV/ion for ΔH_{mix}. In general, a value below -1 meV/ion ensures that the compositions obtained are sufficiently stable as a lower ΔH_{mix} describes the tendency of forming a stable mixed solution [44].

We utilized a batch size of $k = 5$ suggestions for FDI-BO and TOPK-BO, and it was selected by balancing the availability of compute nodes as well as the quality of the batch suggestions; usually, the quality will decrease as the horizon for suggestions is increased. For each methodology, we ran a total of 25 trials, to obtain an average and eliminate the sensitivity to stochastic samplers BO workflow uses.

The results were obtained by profiling the trials in the HPC cluster to obtain the CPU time for each batch cycle

and their corresponding DFT results, as well as job scheduling overhead. One of the reasons we report the time frame in batch cycles instead of CPU time is to eliminate stochastic variations introduced by scheduling overhead and CPU load.

Relative Overall Performance (ROP)

To address the acceleration of the optimization methodology, we have to consider the optimality of the solution it provides as well as its time frame. The optimality criteria of the solution guarantee that the material obtained has the best possible target properties. However, most experiments are resource constrained, i.e., optimization algorithms cannot be run indefinitely, or perhaps an optimal solution may not be trivial to find. We propose a criterion that considers the best value found in the given experimental trial instead, and the time frame required to obtain it. Moreover, each experimental trial might find a different best value at a different time frame than some reference it is being compared with. This motivated us to design a single performance indicator to jointly consolidate both the best values and time frames of the target and reference methodology and output a single relative overall performance value using Eq. (3–6).

$$\mathrm{ROP}_{(\alpha,\beta)} = \alpha \frac{\mathrm{Val(a)} \times I}{\mathrm{Val(ref)}} + \beta \frac{\mathrm{Bat(a)}}{\mathrm{Bat(ref)}} \tag{3}$$

where

$\mathrm{Val} = |\text{Best value reached}|$

$\mathrm{Bat} = \text{Batch cycle to reach the best value}$

$$I = \begin{cases} 1 \text{if} & \mathrm{Val} < -1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\alpha \in (0, 1) : \text{ weight to bias target values} \tag{5}$$

$$\beta = (1 - \alpha) : \text{ weight to bias batch cycles} \tag{6}$$

(a) = FDI-BO, TOPK-BO, or RS as a target, while, (Ref) = S-BO as a reference. α is the importance weight assigned to the best value reached and β is the corresponding importance weight assigned to the batch cycle it takes to reach it. The balancing choice for an α, β is dependent on an individual's design criteria. Note that for any choice of α and β, applying the metric to a reference approach would always yield a value of 1. Moreover, we also employ an indicator function to penalize those values that are above some threshold. Here and in the subsequent section, this threshold is defined at −1meV/ion

Analogous to real-world objectives where time to search for desirable materials is critical, in Eq. 3–6 if β is higher, we bias toward time-factor (acceleration) and if α is higher we bias toward the quality of the best value, making the ROP adaptable to given objectives.

Computational Constraint for the Trials

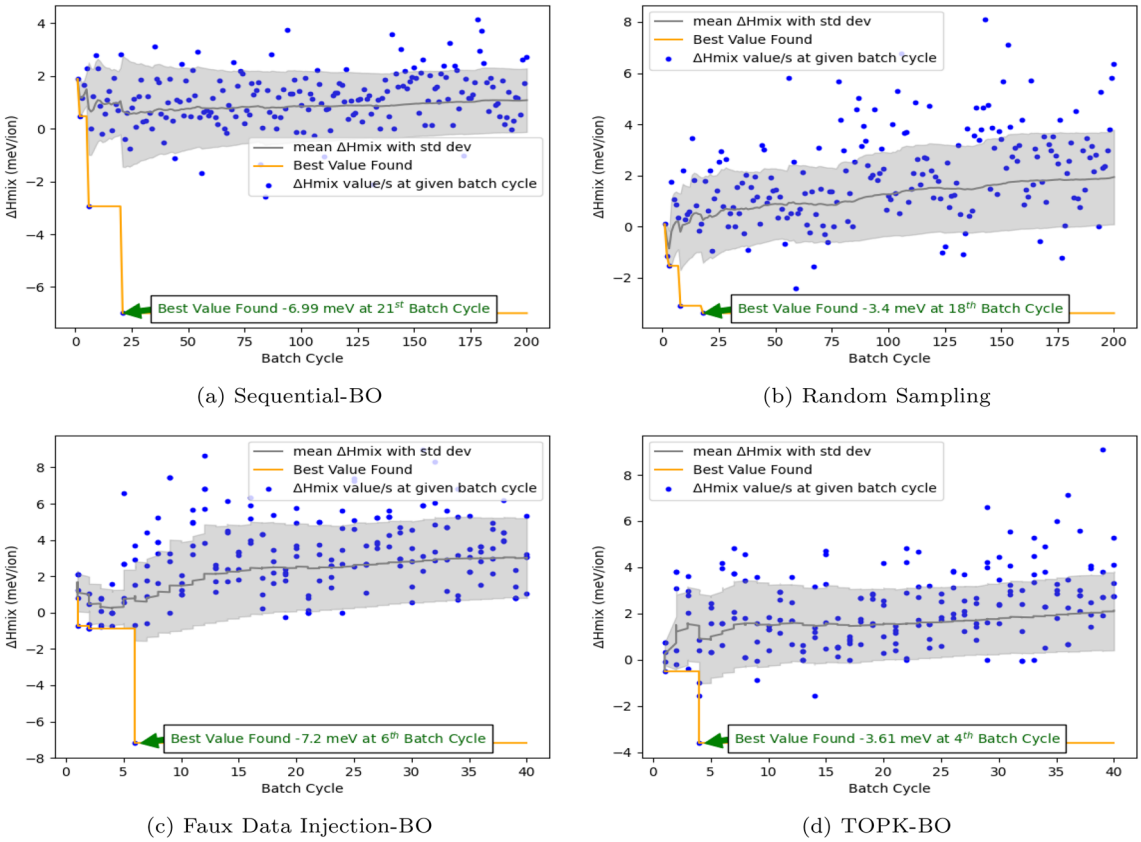
To judge the performance of FDI-BO with other methodologies, we ran 25 computational experiments for each method with a fixed number of DFT evaluations as our computational constraint. As most of the research eco-system is resource constrained, we used a 100 DFT evaluation budget to stop the trials once they reach that mark. Such limitation is well suited in scenarios where we have either limited availability of precursors so we can only prepare a fixed number of samples to test, or only a fixed number of possible configurations of the material that we can evaluate, such as for 2×2×2 supercell of $MA_{0.5}EA_{0.5}PBI_3$, we have only ${}^8C_4 = 70$ configurations to evaluate. Therefore, we subjected our trials to such computational constraints and obtained corresponding best values, computation profiles, and other relevant information for our analysis. In theory, a batch approach such as FDI-BO would execute the experiments faster than a sequential approach as they would utilize k-evaluations in a single batch cycle while S-BO and RS would utilize only 1 sample; however, to analyze if such acceleration affects the quality of the target values we apply ROP metric to the results. Moreover, for the parallel approaches, the completion time is predominantly dependent on the size of the batch, we thus kept the batch size constant at 5, throughout the whole experiment. This choice of batch size is justified in Sect. 4.5.

Selecting the Batch Size

One of the advantages of running batch optimization is that several DFT evaluations can be done at the same time. However, the benefit of such parallelization is only achieved if the suggestions in the batch are informative. In theory, the principle of diminishing return (submodularity) [31] could lead to plateauing of return if the batch size is increased. Therefore, in principle, both FDI-BO and TOPK-BO could become less informative as the size of the batch increases, a similar study done by Gonzales et.al [19] shows that rewards obtained tend to diminish after the batch of size 5. This study in conjunction with the availability of time and computational resources motivated us to run all the parallel computations with the batch of size 5.

Results and Discussion

We ran 25 trials for each methodology to obtain average performance and to discount any outlier introduced due to random seeds of the optimization methods. The computation



(a) Sequential-BO (b) Random Sampling

(c) Faux Data Injection-BO (d) TOPK-BO

Fig. 5 Individual trial examples for each of the considered methods showing suggestions in blue scatter, best value found, and mean/std. dev. of the suggestions. Note that there are five parallel scatter at each batch cycle for batch trials

profiles for the data indicating CPU start and end timings for each DFT computation and the corresponding ΔH_{mix} values are presented in Figs. S2–S5 in the Supplementary section. For each trial, the start time and the end time of DFT computations are represented by rectangular bars whose height corresponds to the output value, i.e., ΔH_{mix}. The overlapping bars in FDI-BO or TOPK-BO indicate parallel computations running in individual nodes of the HPC, while the bars for S-BO are non-overlapping indicating sequential execution. The time where the best value of the trial is indicated by the cyan dashed line. We noticed that the job scheduler had put certain trials on hold in between their execution cycle as can be seen in the case of most FDI-BO trials. This meant that the bare CPU time would not be appropriate to compare individual trails with each other. This is one of the motivations for using batch cycles for our ROP and AF performance indicator metrics. Moreover, comparison in batch cycles helps us eliminate the sensitivity of CPU performance under variable load on individual DFT computation and thereby enables us to address the target proposition without any undesirable confounds.

From Figs. S2–S5, we observe that S-BO and RS expectedly took longer batch cycles to execute than FDI-BO and TOPK-BO. This is also evident in Fig. 5[1] where example instances of FDI-BO and TOPK-BO conclude their sampling budget in lower batch cycles compared to S-BO and RS instances. The plots also depict the extent of the burden required in finding stable structures as most of the suggestions are above −1 meV/ion threshold. The acceleration provided by FDI-BO and TOPK-BO is also evident in convergence curves in Fig. S1, where we only log the best values obtained so far at the corresponding batch cycle.[2] We note that many instances of S-BO took longer than 20 batch cycles to converge to their corresponding best values, while several instances of RS took longer than 40 batch cycles with the quality of the best value not on par with the other methods, thereby discouraging its complete reliance in the search process. S-BO #21 was the top performer for S-BO trials yielding −6.99 meV/ion at 21st batch cycle. Thus, we

[1] Note that the total number of samples in the figures is 200. This was later chopped to match the corresponding constraints.

[2] For the ease of deciphering, we annotated only those trials that reached below 4x the threshold.

Table 1 Average performance (ROP) under various (α, β) combinations. (1.0, 0.0) prioritize the method with the deepest exploration possible without consideration of computational time. (0.2, 0.8) prioritizes the methodology for a quick-and-shallow exploration

	Optimization method			
(α, β)	FDI-BO	S-BO	TOPK-BO	RS
(1.0, 0.0)	1.04	1	0.97	0.92
(0.8, 0.2)	3.52	1	2.71	0.90
(0.5, 0.5)	4.28	1	3.24	0.87
(0.2, 0.8)	4.55	1	3.42	0.86

utilized it as the reference in the ROP and AF performance indicator metrics.

We also like to point out that the average best values for 25 trials for all the optimization methods were around -3 ± 0.2. This indicated that the random sampling method on average performed nearly as well as BO methodologies given the constraints in our experimental setup. However, none of the random sampling trials reached below −4 meV/ion as shown in Fig. S1 and it ran at a relatively higher cost of execution (no. of batch cycles), which in a real scenario might not be feasible. This indicates that averages of best values alone are not sufficient performance indicators to distinguish which optimization method is cost-effective, indicating the need for accommodating decisive performance indicators to address acceleration and the convergence to the best value and the threshold value.

Analyzing Results Under Relative Overall Performance Criteria

From the best values obtained in Fig. S1 for FDI-BO, we deduce that there is no degradation in performance by building a batch of suggestions using the faux-data injection method. We achieved the best performance of FDI-BO in trial FDI-BO #9, of around −7.2 meV/ion at 6*th* batch cycle indicating the potential of FDI-BO to provide similar quality of results that we achieved in our previous work [46]. The best-performing trial for TOPK-BO was −5.18 meV/ion at 3^{rd} batch cycle in trial TOPK-BO #5. To appropriately consolidate best values and corresponding times to yield a single performance value, we utilized Eqs. (3–6). Giving higher importance to best value and lower to batch cycle, $(\alpha, \beta) = (0.8,0.2)$, we get $\text{ROP}_{(0.8,0.2)}$ of 1.52 for FDI-BO #9 and 1.99 to TOPK-BO #5, indicating that they both performed better than S-BO #21 when we jointly consider the best value and corresponding batch cycle.

As our ROP metric is sensitive to the user's selected α, β values, we analyze the results by using various (α, β) combinations. Note that at $(\alpha, \beta) = (1,0)$ the corresponding ROP is merely the ratio of best values of the target approach such as FDI-BO and TOPK-BO with the best value of the reference chosen, which in our case is S-BO #21.

We obtained the average ROP for 25 trails of each methodology and normalized the results w.r.t S-BO in Table 1. The normalization was performed using S-BO results for each (α, β) combination so that for each row we obtain S-BO as 1. This enables us to deduce on average how much each methodology performs w.r.t to average S-BO trials of a given (α, β) combination.

According to Table 1, at $\text{ROP}_{(1.0,0.0)}$ all methodologies share similar values averaging around the 0.92−1.04 range. This is expected as the average of best values found for all optimization methodology was in the range -3 ± 0.2 mV/ion; hence, normalization w.r.t to S-BO would yield the given range, where FDI-BO is at the higher end of the range while RS at the lower end. In general, a higher ROP would favor the methodology that performs sluggish but achieves higher best values while a lower α value would imply that ROP will favor the methodology that returns quick but shallow results. This is evident in Table 1, whereas the β is increased, we observe FDI-BO and TOP-BO on-average score higher ROP values than S-BO and RS. Depending upon α, β, we can bias performance indicator to jointly convey "subjectively," if the best value and acceleration was moderately or significantly better than that of the corresponding reference approach.

We also note that with any combination of α, β, FDI-BO is consistently outperforming all other methodologies, while RS is consistently underperforming. And as the β is increased FDI-BO score increases moderately, indicating that in some of its trials, best values were found quite early on, improving its score as the importance of batch cycle is increased. We anticipated that for the fixed sample trials, S-BO should have performed slightly better than FDI-BO at $\text{ROP}_{(1,0)}$ as S-BO resourcefully utilizes each of the available samples in fixed sample trials, while FDI-BO exhausts 5 by obtaining suggestion based on faux-information. However, this was not the case under the constraints of our experimental design.

Analyzing Results under Acceleration Factor Criteria

The ROP metric focuses more or less on the best values found during the trials. However, for certain use cases, we may only be interested in selecting the methodology that merely reaches the given threshold value. We, therefore, utilized Eq. 2 that considers how fast a given methodology reaches threshold value compared to some reference. Just like ROP, we tabulated the average AF score for each of the optimization methods for threshold values of −1, −2, and −3 meV/ion and normalized w.r.t S-BO in Table 2.

As discussed, AF only accounts for the batch cycle whenever the first instance of threshold value gets reached or

Table 2 Average acceleration factor (AF) using different thresholds for $\Delta Hmix$

Threshold (meV/ion)	Optimization method			
	FDI-BO	S-BO	TOPK-BO	RS
−1	1.84	1	2.07	0.44
−2	1.82	1	1.60	0.39
−3	5.96	1	2.52	0.62

crossed regardless of what the best value in the trial was. This is then compared with the corresponding batch cycle of the reference approach which crosses the same threshold. We used S-BO #21 as the reference trial for AF as well. Both, FDI-BO and TOPK-BO, optimization methods, are likely to outperform S-BO and RS, as running five samples in a given batch cycle increases the probability of crossing the threshold value compared to running only 1 sample. The actual probability advantage an FDI-BO and TOPK-BO have over S-BO is out of scope for this work as for FDI-BO it distinctly depends on the amount of information in the Gaussian process surrogate model, while for TOPK-BO it distinctly depends on the shape/modality of the acquisition function, during batch forming process of the two methods. The probability advantage of both FDI-BO and TOPK-BO is, however, greater than that of S-BO as the first suggestion of both FDI-BO and TOPK-BO in the batch forming process is the same as what an S-BO would propose, assuming the same starting conditions.

Consequently, in Table 2, we observe that both FDI-BO and TOPK-BO on average are able to cross the given threshold faster than S-BO. However, for the -1 meV/ion threshold, TOPK on average outperformed FDI-BO as per the constraints of our experimental design. This was not evident in ROP calculations as ROP only utilizes the corresponding batch cycle at the best value found.

We also note that as the threshold is increased TOPK-BO struggles to cross the threshold values in most of its trials compared to FDI-BO, enabling FDI-BO to score higher in the case of −2 and −3 meV/ion thresholds. This means that FDI-BO has slightly more probability advantage than TOPK-BO in crossing the threshold values due to the quality of its suggestions, in our setup.

We can also look at the distribution of the suggestions of TOPK-BO in Fig. 5d which shows a relatively narrow standard deviation band compared to FDI-BO in Fig. 5c. The naive suggestions of TOPK-BO tend to cluster toward the optimal value provided by the acquisition function in BO while FDI-BO mimics the behavior of S-BO by injecting a faux-data-point for obtaining the next suggestion, and matches the exploration/exploitation behavior of S-BO that is configured with same hyperparameters. We observe roughly 2x acceleration by FDI-BO compared with TOPK-BO for the threshold of −3 and around 6x acceleration compared to S-BO.

Conclusion

In this work, we provided a faux-data injection approach to accelerate the data-driven discovery of materials with the organic–cation halide perovskite as a use case. We compared the faux-data injection method using two performance indicators against various baseline optimization methods such as random sampling, sequential Bayesian optimization, and TOPK Bayesian optimization. We also addressed the importance of the performance indicators in evaluating the quality and acceleration of various optimization methods for various target applications. The two performance indicators that we utilized were; the acceleration factor which helped us determine the acceleration of FDI-BO in finding the material composition that reaches a given threshold value, and relative overall performance; which helped us determine the acceleration of FDI-BO in finding the best material composition.

Based on our analysis, we concluded that the faux-data injection method indeed helped us accelerate our workflow for data-driven discovery of stable halide perovskites for both best value and threshold value criteria. We also showed that if we ignore the time contribution then all given methods on average achieve similar best values based on 100 DFT evaluation constraints in 25 trials. Application of AF and ROP to the results enables us to elicit the effect of acceleration in FDI-BO and TOPK-BO in achieving acceptable best values and threshold values. FDI-BO in general outperformed all the other optimization methodologies due to its quality of suggestions as well as the inclusion of ϵ-greedy method to allow for random suggestions and potentially unbias some suggestions based on faux-data injection loop of FDI-BO. We evaluated the performance using two different criteria. In ROP with $\alpha = 0.2$, FDI-BO outperformed TOPK-BO by a factor of about 1.3 and S-BO by a factor of around 4. In AF, roughly 6x acceleration was obtained by FDI-BO compared to S-BO with a −3 meV/ion threshold, making FDI-BO suitable for accelerating data-driven discovery targeting both the best value and the threshold value.

With the proliferation of self-driving labs and other autonomous experimental set-ups, FDI-BO is well posed in parallelizing data-driven material discovery. The study can, however, be further improved by analyzing a strategic combination of various optimization methods such as PSO, which is considered under the scope of our future work.

Future Work

We aim to expand on this work to include other batch methodologies, providing them with the same evaluation treatment to have a more exhaustive comparison. In particular, we aim to include methods of Gonzales et al. [19], Groves et al. [22], and PSO methods.

One of the limitations of FDI-BO is that although its k-evaluations are run asynchronously in the HPC system, the BO loop itself is synchronous and waits for all the evaluations in the batch to finish before running the next faux-data injection loop to provide the next k-suggestions. This limitation has been overcome by the work of George et al. [6] that leverages data from the surrogate model's mean prediction, Thompson sampling, and random sampling to enable asynchronous candidate suggestion in its version of batch Bayesian optimization. This shall also be the focus of our next study to potentially further accelerate the search process.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s40192-023-00301-x.

Acknowledgements This publication was made possible from the funding received for the Project, An Artificial Intelligence Platform for Accelerating Materials Discovery (AIPAM), awarded by the Hamad Bin Khalifa University Vice President Office grant number VPR-TG01-006. The findings herein reflect the work and are solely the responsibility of the author[s]. We also would like to acknowledge Heesoo Park from the University of Oslo for assisting and troubleshooting the workflow used for Bayesian optimization on organic–cation halide perovskite, Alex Dunn from the University of California for providing the support to implement FDI-BO in *Rocketsled* [10] library, and Anurag Shrivastava from Qatar Computing Research Institute for his assistance in profiling our computations in HPC.

Funding Open Access funding provided by the Qatar National Library.

Code Availability To ensure reproducibility of the results, the code has been made available in our Github repository at https://github.com/elfedwa/FDI-BO

Declarations

Conflict of interest Authors declare no conflict of interests.

References

1. Abatal M, Ruiz-Salvador AR, Hernández NC (2020) A DFT-based simulated annealing method for the optimization of global energy in zeolite framework systems: Application to natrolite, chabazite and clinoptilolite. Microporous Mesoporous Mater 294:109885
2. Abolhasani M, Kumacheva E (2023) The rise of self-driving labs in chemical and materials sciences. Nat Synth 30:1–10
3. Blöchl PE (1994) Projector augmented-wave method. Phys Rev B 50(24):17953
4. Chakraborti N (2004) Genetic algorithms in materials design and processing. Int Mater Rev 49(3–4):246–260
5. Chang C, Lee Y, Wu S (1990) Optimization of a thin-film multilayer design by use of the generalized simulated-annealing method. Opt Lett 15(11):595–597
6. De Ath G, Everson RM, Fieldsend JE (2021) Asynchronous ϵ -greedy Bayesian optimisation. In: Uncertainty in artificial intelligence. PMLR, pp 578–588
7. Di Caro GA, Yousaf AWZ (2021) Multi-robot informative path planning using a leader-follower architecture. In: 2021 IEEE international conference on robotics and automation (ICRA). IEEE, pp 10045–10051
8. Di Caro GA, Ziaullah Yousaf AW (2021) Map learning via adaptive region-based sampling in multi-robot systems. In: International symposium distributed autonomous robotic systems. Springer, pp 335–348
9. Dong Y, Wu C, Zhang C, Liu Y, Cheng J, Lin J (2019) Bandgap prediction by deep learning in configurationally hybridized graphene and boron nitride. npj Comput Mater 5(1):1–8
10. Dunn A, Brenneck J, Jain A (2019) Rocketsled: a software library for optimizing high-throughput computational searches. J Phys Mater 2(3):034002. https://doi.org/10.1088/2515-7639/ab0c3d
11. Faber F, Lindmaa A, von Lilienfeld OA, Armiento R (2015) Crystal structure representations for machine learning models of formation energies. Int J Quantum Chem 115(16):1094–1101
12. Faber FA, Hutchison L, Huang B, Gilmer J, Schoenholz SS, Dahl GE, Vinyals O, Kearnes S, Riley PF, von Lilienfeld OA (2017) Machine learning prediction errors better than DFT accuracy. arXiv preprint arXiv:1702.05532
13. Faber FA, Hutchison L, Huang B, Gilmer J, Schoenholz SS, Dahl GE, Vinyals O, Kearnes S, Riley PF, Von Lilienfeld OA (2017) Prediction errors of molecular machine learning models lower than hybrid DFT error. J Chem Theory Comput 13(11):5255–5264
14. Frazier PI, Wang J (2016) Bayesian optimization for materials design. In: Information science for materials discovery and design, Springer, pp 45–75
15. Gao C, Yang X, Jiang M, Chen L, Chen ZW, Singh CV (2022) Machine learning-enabled band gap prediction of monolayer transition metal chalcogenide alloys. Phys Chem Chem Phys 24(7):4653–65
16. Ghiringhelli LM, Vybiral J, Levchenko SV, Draxl C, Scheffler M (2015) Big data of materials science: critical role of the descriptor. Phys Rev Lett 114(10):105503
17. Ginsbourger D, Le Riche R, Carraro L (2008) A multi-points criterion for deterministic parallel global optimization based on gaussian processes. Tech. rep., HAL, https://hal.archives-ouvertes.fr/hal-00260579
18. Ginsbourger D, Riche RL, Carraro L (2010) Kriging is well-suited to parallelize optimization. In: Computational intelligence in expensive optimization problems, Springer, pp 131–162
19. González J, Dai Z, Hennig P, Lawrence N (2016) Batch bayesian optimization via local penalization. In: Artificial intelligence and statistics, PMLR, pp 648–657

20. Gou J, Lei YX, Guo WP, Wang C, Cai YQ, Luo W (2017) A novel improved particle swarm optimization algorithm based on individual difference evolution. Appl Soft Comput 57:468–481
21. Green MA, Ho-Baillie A, Snaith HJ (2014) The emergence of perovskite solar cells. Nat Photonics 8(7):506–514
22. Groves M, Pyzer-Knapp EO (2018) Efficient and scalable batch bayesian optimization using k-means. https://doi.org/10.48550/ARXIV.1806.01159,https://arxiv.org/abs/1806.01159
23. Hegde G, Bowen RC (2017) Machine-learned approximations to density functional theory hamiltonians. Sci Rep 7(1):1–11
24. Himanen L, Geurts A, Foster AS, Rinke P (2019) Data-driven materials science: status, challenges, and perspectives. Adv Sci 6(21):1900808
25. Huang W, Martin P, Zhuang HL (2019) Machine-learning phase prediction of high-entropy alloys. Acta Mater 169:225–236
26. Ikeda Y (1997) A new method of alloy design using a genetic algorithm and molecular dynamics simulation and its application to nickel-based superalloys. Mater Trans, JIM 38(9):771–779
27. Ingber L (1993) Simulated annealing: Practice versus theory. Math Comput Model 18(11):29–57
28. Jain A, Ong SP, Chen W, Medasani B, Qu X, Kocher M, Brafman M, Petretto G, Rignanese GM, Hautier G, Gunter D, Persson KA (2015) Fireworks: a dynamic workflow system designed for high-throughput applications. Concur Comp-Pract E 27(17):5037–5059. https://doi.org/10.1002/cpe.3505
29. Jansen M (2015) Conceptual inorganic materials discovery-a road map. Adv Mater 27(21):3229–3242
30. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
31. Krause A, Golovin D (2014) Submodular function maximization. Tractability 3:71–104
32. Kresse G, Furthmüller J (1996) Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. Comput Mater Sci 6(1):15–50
33. Kresse G, Furthmüller J (1996) Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. Phys Rev B 54(16):11169–11186
34. Kresse G, Hafner J (1993) Ab initio molecular dynamics for liquid metals. Phys Rev B 47(1):558–561
35. Kresse G, Joubert D (1999) From ultrasoft pseudopotentials to the projector augmented-wave method. Phys Rev B 59(3):1758
36. Liang JJ, Qin AK, Suganthan P, Baskar S (2004) Particle swarm optimization algorithms with novel learning strategies. In: 2004 IEEE international conference on systems, man and cybernetics (IEEE Cat. No. 04CH37583), IEEE, vol. 4, pp 3659–3664
37. Liang Q, Gongora AE, Ren Z, Tiihonen A, Liu Z, Sun S, Deneault JR, Bash D, Mekki-Berrada F, Khan SA et al (2021) Benchmarking the performance of bayesian optimization across multiple experimental materials science domains. npj Comput Mater 7(1):1–10
38. Liu Y, Zhao T, Ju W, Shi S (2017) Materials discovery and design using machine learning. J Mater 3(3):159–177
39. Mao Y, Yang H, Sheng Y, Wang J, Ouyang R, Ye C, Yang J, Zhang W (2021) Prediction and classification of formation energies of binary compounds by machine learning: an approach without crystal structure information. ACS Omega 6(22):14533–14541
40. Matsuoka T, Yamamoto S, Takahara M (2001) Prediction of structures and mechanical properties of composites using a genetic algorithm and finite element method. J Mater Sci 36(1):27–33
41. Monkhorst HJ, Pack JD (1976) Special points for brillouin-zone integrations. Phys Rev B 13(12):5188
42. Ong SP, Richards WD, Jain A, Hautier G, Kocher M, Cholia S, Gunter D, Chevrier VL, Persson KA, Ceder G (2013) Python materials genomics (pymatgen): a robust, open-source python library for materials analysis. Comput Mater Sci 68:314–319
43. Pannetier J, Bassas-Alsina J, Rodriguez-Carvajal J, Caignaert V (1990) Prediction of crystal structures from crystal chemistry rules by simulated annealing. Nature 346(6282):343–345
44. Park H, Ali A, Mall R, Bensmail H, Sanvito S, El-Mellouhi F (2021) Data-driven enhancement of cubic phase stability in mixed-cation perovskites. Mach Learn Sci Technol 2(2):025030
45. Park H, Ali A, Mall R, Bensmail H, Sanvito S, El-Mellouhi F (2021) Data-driven enhancement of cubic phase stability in mixed-cation perovskites. Mach Learn Sci Technol 2(2):025030. https://doi.org/10.1088/2632-2153/abdaf9
46. Park H, Kumar S, Chawla S, El-Mellouhi F (2021) Design principles of large cation incorporation in halide perovskites. Molecules 26(20):6184
47. Peng J, Schwalbe-Koda D, Akkiraju K, Xie T, Giordano L, Yu Y, Eom CJ, Lunger JR, Zheng DJ, Rao RR, Muy S, Grossman JC, Reuter K, Gómez-Bombarelli R, Shao-Horn Y (2022) Human- and machine-centred designs of molecules and materials for sustainability and decarbonization. Nat Rev Mater. https://doi.org/10.1038/s41578-022-00466-5
48. Perdew JP, Burke K, Wang Y (1996) Generalized gradient approximation for the exchange-correlation hole of a many-electron system. Phys Rev B 54(23):16533–16539
49. Perdew JP, Burke K, Ernzerhof M (1996) Generalized gradient approximation made simple. Phys Rev Lett 77:3865–3868
50. Pyzer-Knapp EO, Pitera JW, Staar PW, Takeda S, Laino T, Sanders DP, Sexton J, Smith JR, Curioni A (2022) Accelerating materials discovery using artificial intelligence, high performance computing and robotics. npj Comput Mater 8(1):1–9
51. Saleh E, Tarawneh A, Naser M, Abedi M, Almasabha G (2022) You only design once (yodo): Gaussian process-batch Bayesian optimization framework for mixture design of ultra high performance concrete. Constr Build Mater 330:127270
52. Settles B (2009) Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, USA
53. Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N (2015) Taking the human out of the loop: a review of Bayesian optimization. Proc IEEE 104(1):148–175
54. Son KH, Singh SP, Sohn KS (2012) Discovery of novel phosphors for use in light emitting diodes using heuristics optimization-assisted combinatorial chemistry. J Mater Chem 22(17):8505–8511
55. Takahashi K, Takahashi L (2019) Creating machine learning-driven material recipes based on crystal structure. J Phys Chem Lett 10(2):283–288
56. Tani L, Veelken C (2022) Comparison of bayesian and particle swarm algorithms for hyperparameter optimisation in machine learning applications in high energy physics. arXiv preprint arXiv:2201.06809
57. Tao Q, Xu P, Li M, Lu W (2021) Machine learning for perovskite materials design and discovery. npj Comput Mater 7(1):1–18
58. Tkatchenko A, Scheffler M (2009) Accurate molecular van der Waals interactions from ground-state electron density and free-atom reference data. Phys Rev Lett 102:073005
59. Walsh A (2015) Principles of chemical bonding and band gap engineering in hybrid organic-inorganic halide perovskites. J Phys Chem C 119(11):5755–5760
60. Wang J, Clark SC, Liu E, Frazier PI (2020) Parallel bayesian global optimization of expensive functions. Oper Res 68(6):1850–1865
61. Wang X, Faizan M, Na G, He X, Fu Y, Zhang L (2020) Discovery of new polymorphs of gallium oxides with particle swarm optimization-based structure searches. Adv Electr Mater 6(6):2000119
62. Wilson SR, Cui W (1990) Applications of simulated annealing to peptides. Biopolym Orig Res Biomol 29(1):225–235

63. Yin WJ, Yang JH, Kang J, Yan Y, Wei SH (2015) Halide perovskite materials for solar cells: a theoretical review. J Mater Chem A 3(17):8926–8942

64. Zhuo Y, Mansouri Tehrani A, Brgoch J (2018) Predicting the band gaps of inorganic solids by machine learning. J Phys Chem Lett 9(7):1668–1673

Authors and Affiliations

Abdul Wahab Ziaullah[1] · Sanjay Chawla[2] · Fedwa El-Mellouhi[1]

✉ Fedwa El-Mellouhi
felmellouhi@hbku.edu.qa

Abdul Wahab Ziaullah
awahab@hbku.edu.qa

Sanjay Chawla
schawla@hbku.edu.qa

[1] Qatar Environment and Energy Research Institute, Hamad Bin Khalifa University, 34110 Doha, Qatar

[2] Qatar Computing Research Institute, Hamad Bin Khalifa University, 34110 Doha, Qatar