Bayesian optimization of nanoporous materials

Aryan Deshwal[1], Cory M. Simon[‡2], and Janardhan Rao Doppa[†1]

[1]School of Electrical Engineering and Computer Science, Washington State University. Pullman, WA.
[2]School of Chemical, Biological, and Environmental Engineering. Oregon State University. Corvallis, OR.
[†]jana.doppa@wsu.edu, [‡]cory.simon@oregonstate.edu

June 23, 2021

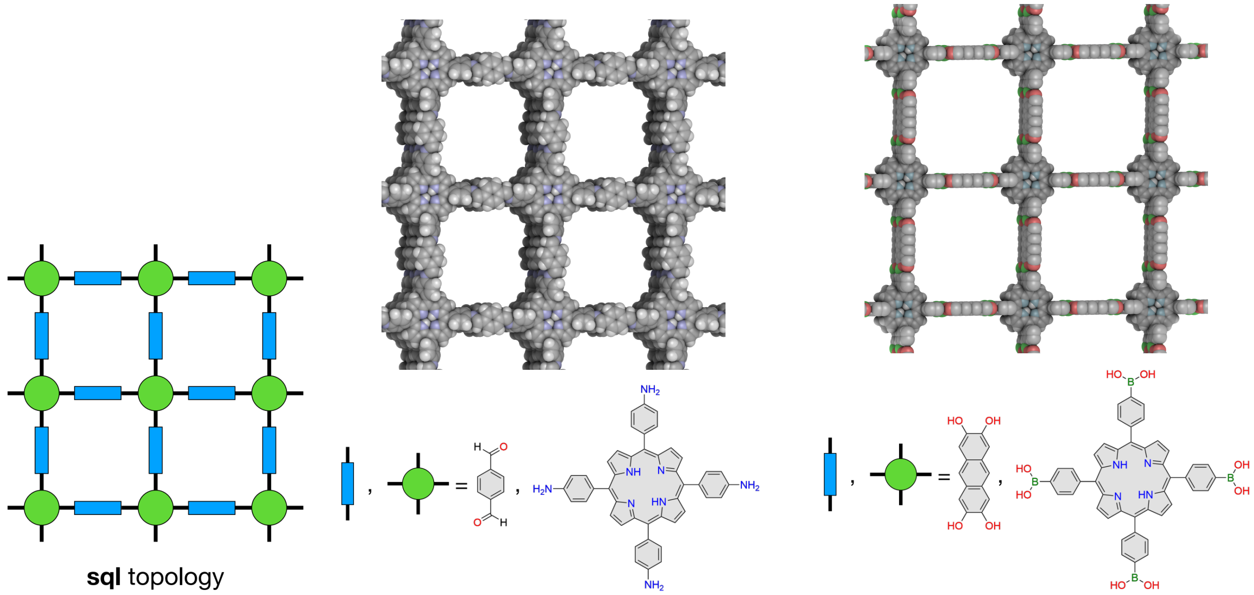
Abstract

Nanoporous materials (NPMs) could be used to store, capture, and sense many different gases. Given an adsorption task, we often wish to search a library of NPMs for the one with the optimal adsorption property. The high cost of NPM synthesis and gas adsorption measurements, whether these experiments are in the lab or in a simulation, often precludes exhaustive search.

We explain, demonstrate, and advocate Bayesian optimization (BO) to actively search for the optimal NPM in a library of NPMs– and find it using the fewest experiments. The two ingredients of BO are a surrogate model and an acquisition function. The surrogate model is a probabilistic model reflecting our beliefs about the NPM-structure–property relationship based on observations from past experiments. The acquisition function uses the surrogate model to score each NPM according to the utility of picking it for the next experiment. It balances two competing goals: (a) exploitation of our current approximation of the structure-property relationship to pick the highest-performing NPM, and (b) exploration of blind spots in the NPM space to pick an NPM we are uncertain about, to improve our approximation of the structure-property relationship. We demonstrate BO by searching an open database of $\sim$ 70 000 hypothetical covalent organic frameworks (COFs) for the COF with the highest simulated methane deliverable capacity. BO finds the optimal COF and acquires 25% of the top 100 highest-ranked COFs after evaluating only $\sim$110 COFs. More, BO searches more efficiently than evolutionary and one-shot supervised machine learning approaches.

1 Introduction

The selective gas adsorption properties of nanoporous materials (NPMs) endow them with many possible applications in the storage [1, 2], separation [3], and sensing [4] of gases. As examples, promising applications of NPMs include (i, storage) densifying hydrogen (H_2)—a clean fuel—for compact storage onboard vehicles [2, 5], (ii, separation) capturing carbon dioxide from flue gas of coal-fired power plants; subsequently sequester it to prevent global warming [6], and (iii, sensing) detecting toxic compounds and explosives [7, 8].



(a)

(b) COF-366

(c) COF-66

Figure 1: Illustrating the modular synthesis of covalent organic frameworks (COFs) [9–11]. (a) The **sql** (square lattice) network topology specifies the connectivity of tetratopic (green) and diptopic (blue) building units. (b, c) Two examples of COFs in the **sql** topology are (b) COF-366 and (c) COF-66 [12]. (top) the crystal structures. (bottom) the building blocks: a planar, tetratopic building unit and a linear, diptopic building unit. The building units are stitched together with covalent bonds, through a condensation reaction, to form 2D COF sheets in the **sql** topology. The sheets stack into layers to form 3D channels. Owing to the rational design and modular synthesis of COFs, on the order of one hundred COFs have been experimentally synthesized and reported [13].

Several classes of NPMs, such as metal-organic frameworks (MOFs) [14], metal-organic polyhedra (MOPs) [15], covalent organic frameworks (COFs) [9], and porous organic cages (POCs) [16], are synthesized modularly by stitching together molecular building blocks via coordination (MOFs, MOPs) or covalent (COFs, POCs) bonds to form $\sim$ crystalline (MOFs, COFs) or molecular (MOPs, POCs) materials. Fig. 1 illustrates the modular synthesis and rational design of COFs in the **sql** topology [12]. The many topologies [9, 17, 18], abundance of molecular building blocks, and post-synthetic modifiability [19, 20] permit an unlimited number of possible structures exhibiting diverse adsorption properties.

A common goal is to find, among a large set of candidate NPM structures, the NPM structure(s) with the optimal adsorption property for a given application. As opposed to an exhaustive search, our goal is to search for the optimal NPM *efficiently*, by consuming minimal resources (computational and/or physical) in the process. In the laboratory setting, synthesizing an NPM and measuring its property costs labor and raw materials, and throughput is limited by the capital equipment in the lab. In the computational setting, constructing a high-fidelity computational model of an NPM structure [21–25] and predicting its gas adsorption property through molecular simulations [26, 27] consumes electricity, and throughput is limited by computing resources. Thus, both in the bona fide laboratory and on the computer, our goal is to find the optimal NPM(s) for a given adsorption task using the fewest experiments (experiment = constructing an NPM and evaluating its adsorption property).

In this article, we explain, demonstrate, and advocate Bayesian optimization (BO) to *actively* and *efficiently* search for NPMs with an optimal property for a given adsorption task. Active, because each BO iteration performs three consecutive steps: (a) conducting an experiment on an NPM, (b) updating our belief about the structure-property relationship, and (c) selecting the NPM for the next experiment. See Fig. 2 for a high-level illustration. Efficient, because BO makes a *data-informed* decision on the NPM to select for the next experiment, while balancing: (a) *exploitation* of our current (uncertain!) data-driven belief about the structure-property relationship to pick an NPM that might have the optimal property and (b) *exploration* of regions of the NPM design space where our belief about the structure-property relationship is weak to pick an NPM we are uncertain about.

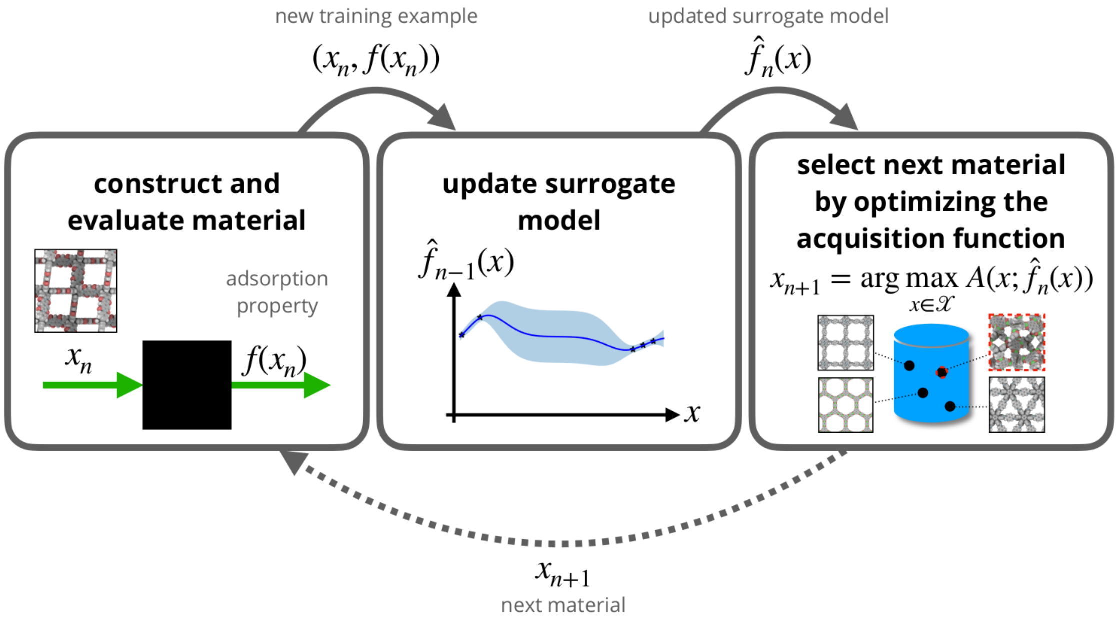


Figure 2: Illustration of a single iteration of active search in Bayesian optimization (BO) of nanoporous materials. First, we conduct an experiment that measures the property $f(x_n)$ of the material represented by x_n. Second, using the new observation, we update our belief about the underlying objective function $f(x)$, reflected by the surrogate model. Third, we use the acquisition function to pick the next material for an experiment. BO is an active search method to find the material x that optimizes the black-box function $f(x)$.

The two key components of BO are a surrogate model and an acquisition function. The surrogate model, with "surrogate" hinting at "substitute for the experiment"– is a probabilistic model for the structure-property relationship. It is trained on all observations from past experiments. The surrogate model cheaply predicts the properties of the unevaluated NPMs and, critically, quantifies the uncertainty in its predictions. The acquisition function is used to make the decision of which NPM to select for the next experiment. It uses the surrogate model to score the utility of selecting each unevaluated NPM for the next experiment by striking a balance in the exploitation-exploration trade-off. The acquisition function is maximal in regions of the NPM design space where (i) we believe high-performing materials will reside and/or (ii) we are uncertain about the structure-property relationship. The acquisition function relies on the surrogate model to give a resource-efficient, intelligent, active search for optimal NPMs in a wide variety of contexts.

We demonstrate BO of NPMs by efficiently searching an open database [28] of $\sim$70 000 hypothetical COFs, represented by vectors of hand-designed features based on domain knowledge, for those with the highest simulated methane deliverable capacity to store natural gas onboard vehicles [2]. BO recovers the optimal (COFs) using fewer experiments than incumbent strategies including random search, evolutionary algorithms, and one-shot supervised machine learning. In the outlook, we discuss several active research areas in BO that are likely to apply to several problems in NPM discovery: (i) batch BO where experiments are parallelized, (ii) multi-fidelity BO, where NPMs can be evaluated using multiple methods which vary in accuracy and resource cost, (iii) multi-objective BO, where we aim to find a Pareto optimal set of NPMs to optimize multiple properties, and (iv) constrained BO, where our goal is to find high-performing NPMs which can be synthesized.

2 Review of previously used NPM search methods

The NPM research community has adopted several approaches to efficiently search a library of NPMs for the optimal NPM(s) [29, 30]. We define the efficiency of a search strategy with respect to two naive baselines: (i) exhaustive search, where we conduct the high-fidelity experiment on every NPM in the library, and (ii) random search, where we conduct the high-fidelity experiment on a (uniform) random sample of the NPMs in the library.

Supervised machine learning models. A supervised machine learning model can serve as a cheaper, albeit lower fidelity, surrogate for the high-fidelity experiment [29, 31–35], thereby reducing the cost of an exhaustive search. A machine learning approach is predicated upon cheaply computed (relative to the experiment) vector representations of the NPMs– hand-engineered [36, 37] or learned [38]– that encode structural features and are correlated to the property. Training examples are gathered by selecting a small (random or diverse [39, 40]) subset of the library of NPMs and labeling them with the property values using the high-fidelity experiment. Using the training examples, the supervised machine learning model learns to predict the property of any given NPM from its vector representation. The trained model is then used, as a surrogate for the high-fidelity experiment, to cheaply predict, from their vector representations, the properties of the remaining NPMs in the library. Further high-fidelity experiments may be directed on the NPMs predicted to be optimal by the machine learning model. See Refs. [39, 41–49] as examples.

Genetic algorithms. Genetic algorithms [50] are iterative, stochastic search methods inspired by Darwinian evolution. Each NPM is represented by a "chromosome"– a vector of categorical variables that uniquely specifies its structure. A small initial generation (set) of property-labeled NPMs is iteratively evolved by applying genetic operations to their chromosomes: mutation, replication, selection, and recombination. At each generation, we conduct experiments on each newly evolved NPM to evaluate its fitness. This guides the genetic operations used to evolve to the next generation of chromosomes, with the ambition of both exploring NPM space and enriching future generations with high-fitness NPMs. See Refs. [51–53] as examples.

Monte Carlo tree search. When the NPM search space can be framed as a tree, Monte Carlo tree search [54] is more efficient than random search. Starting at the root node, a policy to select a child node is iteratively applied, giving a path through the tree, culminating at a leaf node. The experiment is then conducted on the NPM represented by the leaf node. Its measured property, viewed as a reward, is back-propagated through the tree to update the statistics of each node along the path to it. Both the visit counts and reward allocations of the nodes are used in the tree policy to balance exploration of new branches of the tree that have not been visited often (or at all) and exploitation of current knowledge to follow branches of the tree that appear likely to lead to optimal NPMs. See Refs. [55, 56] as examples.

Each of these prior approaches suffer from drawbacks. The supervised machine learning approach passively, in one-shot, selects training data, requiring a large training set to learn a good predictor of the property from the structure representation. Active learning [57] can be used to reduce the number of training examples, but this mechanism is not geared towards finding the optimal NPM using the fewest experiments. Genetic algorithms are sequential, active search procedures aimed at quickly finding the optimal NPM. However, the genetic operations are heuristic and do not balance exploration and exploration rigorously. As a consequence, genetic algorithms can be difficult to tune and could require many experiments to find the optimal NPMs. MCTS balances exploration and exploitation more rigorously. But, it requires many NPM evaluations to identify promising regions of the tree because it does not explicitly leverage the similarity among structures for principled exploitation. Further, MCTS is limited to NPM design spaces which can be framed as a tree.

3 Problem setup: find the optimal material

Suppose we have a large database of candidate NPM structures, $\mathcal{X}$, for some adsorption task. Let $f : \mathcal{X} \rightarrow \mathbb{R}$ be a black-box objective function that, given a candidate NPM $\mathbf{x} \in \mathcal{X}$, returns the relevant adsorption property $y = f(\mathbf{x})$. Each evaluation of f corresponds to performing an *expensive* experiment—either in the laboratory or in a molecular simulation—to measure the adsorption property y of NPM $\mathbf{x}$. Our goal is to find the highest-performing NPM $\mathbf{x}^*$ from $\mathcal{X}$ that maximizes the objective function f,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}), \tag{1}$$

while conducting the fewest number of expensive experiments.

We can interpret $f(\mathbf{x})$ as the [unknown] structure-property relationship [58, 59] since $\mathbf{x}$ [abstractly, at this point in our discussion] represents the structure of a unique NPM and evaluating f means conducting an experiment to measure its property, y.

4 Overview of Bayesian Optimization

We explain the key ideas behind the Bayesian Optimization (BO) framework [60,61] to find the highest-performing NPM—solve the problem in eqn. 1—*efficiently*, by using the fewest (expensive) experiments.

4.1 Defining an NPM feature space

While $\mathbf{x}$ as an abstract representation of an NPM suffices for defining the problem in eqn. 1, for BO we must concretely define a NPM *feature space* or *search space* in which each NPM $\mathbf{x}$ lies.

Without loss of generality, take $\mathbf{x}$ to be a fixed-size (among all NPMs) vector representation of the NPM that lies in a continuous feature space. The NPM feature vectors $\{\mathbf{x}\}$ should be designed to (1) encode the relevant structural and chemical features of the NPMs, (2) be rotation-, translation-, and, if the NPM is a crystal, replication-invariant, (3) be cheap to compute compared to conducting the experiment (evaluating f), and (4) ideally, each correspond to a unique NPM (mapping NPM $\rightarrow \mathbf{x}$ is injective) [62]. As a result, NPMs close in the feature space should exhibit close values of the property y.

The simplest example of a representation $\mathbf{x}$ of an NPM is a list of hand-designed, based on domain expertise, descriptors/features of its structure and composition, such as pore volume, surface area, largest included sphere diameter, density, weight fraction carbon, etc. [63–65]. Alternatively, $\mathbf{x}$ could be learned from a graph [66–68] or 3D image representation [69, 70] of a NPM by a graph neural network [38] or convolutional neural network [71], respectively. See reviews in Refs. [29, 72] for defining feature vectors of NPMs and Refs. [39, 41, 43, 73–76] for different examples of NPM feature spaces. As opposed to dwelling on how to *define* a good feature space of NPMs, we will instead focus on BO, a technique to *search* the defined feature space for the optimal NPM $\mathbf{x}^*$ in an efficient manner.

4.2 Active search: exploitation vs. exploration

Even with an NPM feature space defined, in practice, the structure-property relationship $f(\mathbf{x})$ is a black-box function; analytical expressions for $f(\mathbf{x})$ and/or its gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$ are not known, and it may be multi-modal.

BO is a derivative-free method to *actively* and *efficiently* search the database of NPMs $\mathcal{X}$ for the NPM $\mathbf{x}^*$ that maximizes $f(\mathbf{x})$. Active, because BO sequentially selects NPMs from $\mathcal{X}$ for experimentation (to evaluate with f), iterating between conducting an experiment and making a decision about which experiment to conduct next. Efficient, because BO makes a data-driven decision to select the next NPM for an experiment while taking into account all observed (NPM $\mathbf{x}$, property y=$f(\mathbf{x})$) pairs from previous experiments. Each decision to select the next unevaluated NPM from $\mathcal{X}$ to evaluate with f must trade-off two conflicting goals:
1) *Exploitation* suggests to use our current, but uncertain, approximation of the structure-property relationship, based on the past observations, to select the NPM that appears to have the most promise as a high-performing material.
2) *Exploration* suggests to select the NPM that we are most uncertain about to improve our approximation of the structure-property relationship.
So, to balance exploitation and exploration, we must balance visits to regions of NPM feature space that (i) appear to contain high-performing NPMs and (ii) have not been explored well. A colloquial example of the exploitation-exploration dilemma in our lives is, in aiming to maximize our enjoyment of food, whether to choose a restaurant that we have visited and know we like versus a new one [77].

4.3 The ingredients of BO for data-driven decision-making: a surrogate model and an acquisition function

In the BO framework, the two key components used to make each sequential decision of which NPM to conduct an experiment on next are (1) a *surrogate model* that captures our beliefs, based on past observations, about the structure-property relationship and (2) an *acquisition function* that scores each NPM according to the utility of conducting the experiment on it next. The acquisition function uses the surrogate model of the structure property-relationship $f(\mathbf{x})$ to decide which NPM to evaluate next while striking a balance between exploitation and exploration.

The surrogate model. The surrogate model is a probabilistic model of the structure-property relationship $f(\mathbf{x})$ trained on all observations[1] $\{(\mathbf{x}_i, y_i{=}f(\mathbf{x}_i))\}_{i=1}^{n}$ from past experiments. Typically, the surrogate model treats $f(\mathbf{x})$ as a random variable that follows a Gaussian distribution:

$$f(\mathbf{x}) \sim \mathcal{N}(\hat{y}(\mathbf{x}), \sigma^2(\mathbf{x})) \tag{2}$$

with mean $\hat{y} \in \mathbb{R}$ and variance $\sigma^2 \in \mathbb{R}$. The surrogate model reflects our current beliefs about $f(\mathbf{x})$ and serves two purposes in BO. First, to guide exploitation, $\hat{y}(\mathbf{x})$ *cheaply* estimates the properties of the remaining, unevaluated NPMs, i.e., $\hat{y}(\mathbf{x})$ is a cheap-to-evaluate approximation of the expensive-to-evaluate objective function $f(\mathbf{x})$. Second, to guide exploration, $\sigma^2(\mathbf{x})$ quantifies the uncertainties in the predicted properties of the unevaluated NPMs. This makes us aware of "blind spots" of the surrogate model—regions in the NPM feature space we need to explore to improve our approximation $\hat{y}(\mathbf{x})$ and reduce the uncertainty in our beliefs about $f(\mathbf{x})$.

The surrogate model is updated in each iteration of BO, after the new observation $(\mathbf{x}_{n+1}, y_{n+1}{=}f(\mathbf{x}_{n+1}))$ is gathered, to (i) improve the approximation of $f(\mathbf{x})$ and (ii) account for the reduced uncertainty in the region of the feature space surrounding the newly evaluated NPM $\mathbf{x}_{n+1}$. Consequently, let us denote the surrogate model after iteration n of BO as $\hat{f}_n : \mathbf{x} \mapsto (\hat{y}, \sigma)$.

The acquisition function. The *acquisition function* $A(\mathbf{x}; \hat{f}(\mathbf{x})) : \mathcal{X} \rightarrow \mathbb{R}$ scores the utility of, next, evaluating NPM $\mathbf{x} \in \mathcal{X}$ with the expensive objective function f. Here, "utility" is defined in terms of our ultimate goal of finding the optimal NPM $\mathbf{x}^*$ in eqn. 1 with the fewest experiments. The acquisition function employs the prediction of the property $\hat{y}$ and the associated uncertainty σ^2 from the surrogate model $\hat{f}(\mathbf{x})$ to assign a utility score to the NPM that balances exploitation and exploration, respectively. Maxima of the acquisition function are located in regions of NPM feature space where the predicted property is large and/or uncertainty is high.

The decision of which NPM to evaluate next is made by maximizing the acquisition function:

$$\mathbf{x}_{n+1} = \operatorname*{arg\,max}_{\mathbf{x} \in \mathcal{X} \setminus \mathcal{X}_n} A(\mathbf{x}; \hat{f}_n(\mathbf{x})), \tag{3}$$

where $\mathcal{X}_n := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ is the acquired set of n NPMs that have been evaluated already. Importantly, the acquisition function must be cheap to evaluate and optimize (if $\mathcal{X}$ is not a finite set).

[1] Without loss of generality, the observations are assumed noise-less for clarity of presentation.

4.4 Summarizing: BO active search iterations

Fig. 2 illustrates an iteration of BO. At the beginning of iteration n, we conduct an experiment on NPM $\mathbf{x}_n$, i.e., we evaluate NPM $\mathbf{x}_n$ with the objective function f to obtain a new observation $(\mathbf{x}_n, y_n = f(\mathbf{x}_n))$. Next, we update the old surrogate model $\hat{f}_{n-1}(\mathbf{x})$ to account for this new observation, giving the new surrogate model $\hat{f}_n(\mathbf{x})$. We then select the next (unevaluated) NPM to evaluate, $\mathbf{x}_{n+1}$, as the one that maximizes the acquisition function $A(\mathbf{x}; \hat{f}_n(\mathbf{x}))$.

We terminate BO after we either (i) expend our budget for experiments or (ii) find a material with a satisfactory property value. The BO solution to the problem in eqn.1, $\mathbf{x}^*$, then follows from the evaluated NPM with the highest observed property, $\arg\max_{i=1}^{N} y_i$, where N is the number of BO iterations (=experiments) performed. We refer the reader to [78] for theoretical guarantees of BO including the assumptions to find the global optima and the quality of the approximate solution as a function of the number of BO iterations.

4.5 Remark: BO vs. active learning

We remark on a distinction between active learning [79] and Bayesian optimization. Both sequentially collect training examples for a supervised machine learning model. In active learning, the examples are efficiently collected with the goal of reducing the uncertainty in the machine learning model. In Bayesian optimization, the examples are efficiently collected with the goal of, instead, finding the optimal material. BO is more efficient for finding the optimal material than active learning because it avoids collecting examples in regions of feature space that contain poor-performing materials, whereas active learning will do so to reduce the uncertainty of the model in those regions.

5 Surrogate models and acquisition functions

In this section, we explain surrogate models and acquisition functions commonly used in BO.

5.1 Surrogate models: Gaussian processes

Gaussian processes (GPs) [80] are the most commonly used surrogate models in BO owing to their flexibility as function approximators, principled uncertainty quantification, and compatibility with the kernel trick. GPs are non-parametric models that can approximate complicated objective functions $f(\mathbf{x})$ given labeled training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$. Through a Bayesian probabilistic framework, GPs provide uncertainty estimates in their predictions and allow incorporation of prior beliefs. GPs rely on a kernel function $k(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ [81] to capture the similarity between any two NPMs $\mathbf{x}$ and $\mathbf{x}'$. This gives GPs the flexibility to approximate arbitrary, complicated (but well-behaved!) functions $f(\mathbf{x})$. Moreover, it gives GPs versatility in how to represent the NPMs, e.g., graph kernels [82] can be used for NPMs represented as crystal graphs (used to predict methane adsorption in MOFs [83]).

What is a GP? A GP is a stochastic process that treats the value of the objective function at any given point in feature space, $f(\mathbf{x})$, as a random variable. Specifically, GPs assume the joint distribution of any finite collection of function values, say at points $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ on its domain, follows a multivariate Gaussian distribution

$$\mathbf{f} \equiv [f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_m)]^\intercal \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \tag{4}$$

whose covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$ is given by the kernel function applied pairwise over the points $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$, $\Sigma_{i,j} := k(\mathbf{x}_i, \mathbf{x}_j)$. The kernel function $k(\mathbf{x}, \mathbf{x}')$ quantifies the similarity of NPMs $\mathbf{x}$ and $\mathbf{x}'$; hence, the idea in GPs is that the properties $f(\mathbf{x})$ and $f(\mathbf{x}')$ of similar (dissimilar) NPMs $\mathbf{x}$ and $\mathbf{x}'$ are highly (un)correlated. [84] GPs effectively model the entire function $f(\mathbf{x})$— in a point-wise manner— by assuming eqn. 4 holds for *any* arbitrary, finite collection of function values on its domain. The mean of zero in eqn. 4 assumes the measurements are centered.

From a Bayesian perspective, eqn. 4 is a prior assumption about the structure-property relationship $f(\mathbf{x})$. When we gather new observations, we will update this prior assumption to arrive at the posterior distribution reflecting our beliefs about the structure-property relationships in light of new data.

Kernel functions. Examples of kernel functions that operate on vector representations of two NPMs $\mathbf{x}$ and $\mathbf{x}'$ include the linear, polynomial, and radial basis function (RBF) kernels:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f \mathbf{x}^\intercal \mathbf{x}' \qquad \text{linear kernel} \tag{5}$$

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f (\mathbf{x}^\intercal \mathbf{x}')^d \qquad \text{homogeneous polynomial kernel} \tag{6}$$

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f e^{-||\mathbf{x}-\mathbf{x}'||_2^2/(2\gamma^2)}. \qquad \text{radial basis function (RBF) kernel} \tag{7}$$

Each kernel possesses the hyperparameter σ_f, the signal variance, which is a scale factor controlling the range of the function values represented by a GP.The polynomial kernel has a hyperparameter d that controls the order of the polynomial in the features, and the RBF kernel contains a length-scale hyperparameter γ that controls how close $\mathbf{x}$ and $\mathbf{x}'$ must be in the feature space to be considered “similar”. Implicitly, each nonlinear kernel maps the two vectors $\mathbf{x}$ and $\mathbf{x}'$ into a new, higher-dimensional feature space through a mapping τ, then takes the inner product of the vectors in the new feature space:

$$k(\mathbf{x}, \mathbf{x}') = \tau(\mathbf{x})^T \tau(\mathbf{x}'). \tag{8}$$

Interestingly, the feature map $\tau(\mathbf{x})$ corresponding to the RBF kernel in eqn. 7 maps vectors into an *infinite* dimensional feature space! By implicitly operating in a higher-dimensional feature space, nonlinear kernels give GPs more flexibility, or expressiveness, for approximating complicated objective functions $f(\mathbf{x})$. Notably, graph kernels [82] can define similarities of two NPMs represented as graphs (nodes: atoms, edges: bonds). Similarly, we can also employ kernels for images [85] by representing MOFs as images.

Inference with GPs. In BO, we exploit GPs for regression, with uncertainty quantification, to build a surrogate model for $f(\mathbf{x})$. We have observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ from previous experiments (previous iterations of BO), with y_i the measured property of NPM $\mathbf{x}_i$. Under the Bayesian view, y_i is a noise-free observation[2] of the random variable $f(\mathbf{x}_i)$. We wish to know the distribution of the random variable $f(\mathbf{x})$ for an *unevaluated* NPM $\mathbf{x}$, to determine the utility of evaluating it in the next experiment. Imposing the assumption in eqn. 4 for a specific collection of points on the domain of $f(\mathbf{x})$ composed of (i) the n evaluated NPMs from the past experiments $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ and (ii) the unevaluated NPM, $\mathbf{x}$:

$$\begin{bmatrix} \mathbf{f} \\ f(\mathbf{x}) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\sigma} \\ \boldsymbol{\sigma}^\intercal & k(\mathbf{x}, \mathbf{x}) \end{bmatrix}\right) \tag{9}$$

with $\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), ..., f(\mathbf{x}_n)]^\intercal$ the vector of random variables representing the properties of the previously evaluated NPMs, $\boldsymbol{\sigma} = [k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), ..., k(\mathbf{x}, \mathbf{x}_n)]^\intercal$ the vector of the kernel between the unevaluted NPM and the previously evaluated NPMs, and $\boldsymbol{\Sigma}$ the kernel matrix for the previously evaluated NPMs with $\Sigma_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) \quad \forall i, j \in \{1, 2, ..., n\}$. However, we have observations of the random variables in $\mathbf{f}$, $\mathbf{y} = [y_1, y_2, \cdots, y_n]^\intercal$. Conditioning the joint distribution in eqn. 9 on the observations $\mathbf{y}$, we arrive at the posterior distribution for the property $f(\mathbf{x})$ of the *unevaluated* NPM, also Gaussian:

$$f(\mathbf{x})|\mathbf{y} \sim \mathcal{N}\left(\hat{y}(\mathbf{x}), \sigma^2(\mathbf{x})\right). \tag{10}$$

with mean and variance:

$$\hat{y}(\mathbf{x}) = \boldsymbol{\sigma}^\intercal \boldsymbol{\Sigma}^{-1} \mathbf{y} \tag{11}$$

$$\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \boldsymbol{\sigma}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{\sigma}. \tag{12}$$

We can interpret eqns. 11 and 12. The predicted property of the unevaluated NPM, $\hat{y}(\mathbf{x})$, is a linear combination of the observed properties of the evaluated NPMs, $\mathbf{y}$ with weights $\boldsymbol{\sigma}^\intercal \boldsymbol{\Sigma}^{-1}$. The weight on each measured property depends on the similarity between that NPM and the unevaluated NPM. The variance $\sigma^2(\mathbf{x})$ describing the uncertainty associated with the prediction of the property of the unevaluated NPM $\mathbf{x}$ is the prior assumption of $k(\mathbf{x}, \mathbf{x})$ reduced by $\boldsymbol{\sigma}^\intercal \boldsymbol{\Sigma}^{-1} \boldsymbol{\sigma}$, which captures the similarity of the unevaluated NPM $\mathbf{x}$ with the set of previously evaluated NPMs.

Fig. 3 illustrates a GP model of a toy function $f(x)$, based on an RBF kernel, over a toy one-dimensional NPM feature space $\mathcal{X}=\mathbb{R}$, trained on five observations. The mean in the GP, $\hat{y}(x)$, approximates $f(x)$, and the variance $\sigma^2(x)$ expresses uncertainty in the approximation. Generally, the uncertainty is small close to an observation and large when far from an observation.

Hyperparameters of GPs. GPs are non-parametric models, but most useful kernel functions used in GPs contain hyper-parameters. For example, the RBF kernel in eqn. 7 has the length-scale γ and the signal variance σ_f hyperparameters. To learn the hyper-parameters of the kernel that give us the best approximation of $f(\mathbf{x})$, we could perform a grid search over hyperparameter space for those that minimize the validation loss. A more popular and faster way to estimate hyperparameters of the kernel is to maximize the marginal likelihood of the observed data as a function of the kernel hyperparameters. [84] Generally, at each iteration of BO, the hyperparameters of the kernel are updated to account for the newly acquired observation.

[2] Note that we can pose GPs that relax the assumption that the observations are noise-free [80].

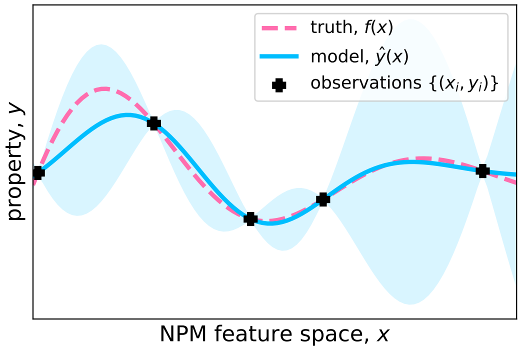


Figure 3: Illustration of a Gaussian process (GP) model with an RBF kernel over a toy one-dimensional NPM feature space. The black points are the observed data from a toy structure-property relationship, $f(x)$. The blue line and shaded region visualize the GP model trained on the observed data: the line is the approximation $\hat{y}(x)$ of the structure-property relationship, while the shaded region illustrates the uncertainty by covering $\hat{y}(x) \pm 2\sigma(x)$. The GP model shows large (small) uncertainty in regions of feature space far from (close to) the observations.

Two further interpretations of GP models of functions. A GP model of the objective function $f(\mathbf{x})$ can be interpreted as (i, weight space view) Bayesian linear regression in the implicit feature space of the kernel and (ii, function space view) a distribution over functions [80]. To clarify, the GP model of $f(\mathbf{x})$ in eqn. 4 is equivalent to the parametric model:

$$f(\mathbf{x}) = \mathbf{w}^\intercal \tau(\mathbf{x}), \tag{13}$$

with weights **w** on which we place a Gaussian prior and τ the map associated with the kernel $k(\mathbf{x}, \mathbf{x}')$ used in the GP. In the weight space view, GP inference models the posterior distribution over weights **w** in eqn. 13. In the function space view, GP inference models the posterior distribution over the space of functions represented in eqn. 13. Though eqn. 13 is helpful for understanding GPs and sampling functions from the distribution over the function space they describe, we in practice conduct GP inference using the kernel, through eqns. 11 and 12. E.g. $\tau(\mathbf{x})$ is a vector of infinite dimension in the case of the RBF kernel, making the view of GPs in eqn. 13 unfriendly for computations.

5.2 Examples of acquisition functions

We provide three common examples of acquisition functions and explain how they use the surrogate model to trade-off exploration and exploitation to select the NPM to evaluate in the next experiment.

Upper confidence bound (UCB). The UCB acquisition function selects the point that maximizes the upper confidence function:

$$A(\mathbf{x}) := \hat{y}(\mathbf{x}) + \beta\sigma(\mathbf{x}) \tag{14}$$

where $\hat{y}(\mathbf{x})$ and $\sigma(\mathbf{x})$ are the predicted property of NPM **x** and its associated uncertainty, respectively, provided by the surrogate model. The parameter β explicitly trades-off exploration and exploitation. If β is large, the UCB tends to select NPMs with the highest uncertainty to explore the feature space. If β is small, the UCB tends to select NPMs with the highest predicted property to exploit our current approximation of the structure-property relationship $f(\mathbf{x})$. To clarify the UCB terminology, the top boundary of the shaded region that bands $\hat{y}(x)$ in Fig. 3 is the UCB for $\beta = 2$.

Expected improvement (EI). Another acquisition strategy is to select the NPM with the highest expected improvement (EI) of the property of the best evaluated NPM so far. Let the random variable $I(\mathbf{x})$ = $\max(0, f(\mathbf{x}) - \max_i y_i)$ denote the improvement in the property of a NPM **x** over the best observed NPM thus far. The EI is then:

$$A(\mathbf{x}) := \int_{-\infty}^{\infty} I(\mathbf{x})\mathcal{N}(y|\hat{y}(\mathbf{x}), \sigma^2(\mathbf{x}))dy, \tag{15}$$

which can be written in closed form:

$$A(\mathbf{x}) = \begin{cases} (\hat{y}(\mathbf{x}) - \max_i y_i)\Phi\left(\frac{\hat{y}(\mathbf{x}) - \max_i y_i}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\phi\left(\frac{\hat{y}(\mathbf{x}) - \max_i y_i}{\sigma(\mathbf{x})}\right) & \sigma(\mathbf{x}) > 0 \\ 0 & \sigma(\mathbf{x}) = 0, \end{cases} \tag{16}$$

with Φ and ϕ the cumulative and probability distribution functions, respectively, of the standard normal distribution. The first and second terms in eqn. 16, respectively, capture the exploitation and exploration component of EI.

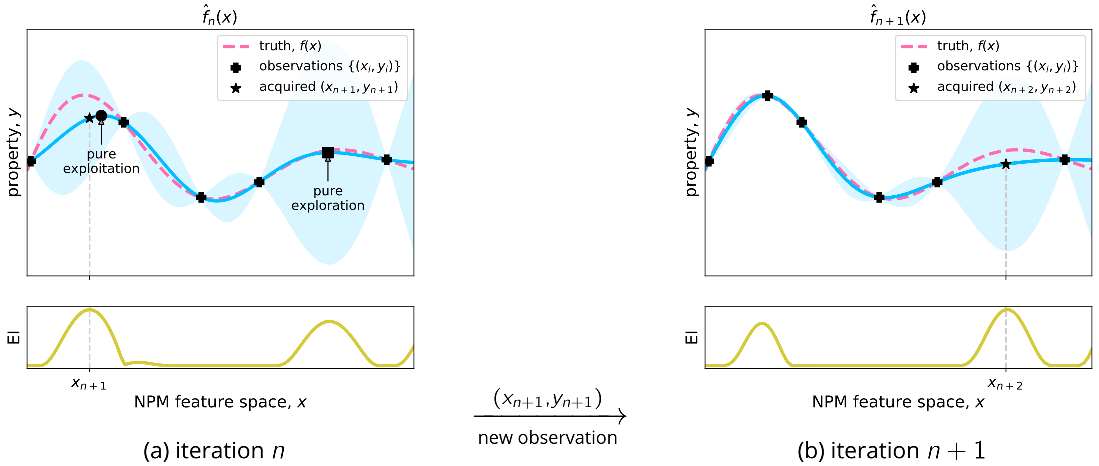
Information-theoretic acquisition functions. The principle behind acquisition functions based on information theory is to select the NPM **x** that maximizes the mutual information between (i) the observation of its property y=$f(\mathbf{x})$ and (ii) the location of the NPM $\mathbf{x}^*$ in feature space that maximizes $f(\mathbf{x})$. Viewing both $f(\mathbf{x})$ and $\mathbf{x}^*$ as random variables, the following acquisition function describes the mutual information between the observation $(\mathbf{x}, y$=$f(\mathbf{x}))$ of the property of a newly acquired NPM, **x**, and the location of the optimal NPM, $\mathbf{x}^*$.

$$A(\mathbf{x}) = IG[(\mathbf{x}, y), \mathbf{x}^*] \tag{17}$$

$$= H[p(\mathbf{x}^*)] - H[p(\mathbf{x}^*|(\mathbf{x}, y))] \tag{18}$$

where $IG[\cdot, \cdot]$ is the mutual information between two random variables and $H(\cdot)$ is the entropy of a probability distribution. The mutual information is the reduction in the entropy of the probability density function of the location of the optimum NPM, $\mathbf{x}^*$, as a result of observing the property y=$f(\mathbf{x})$ of NPM **x** and thus conditioned on knowing $(\mathbf{x}, y)$. Conceptually, the distribution $p(\mathbf{x}^*)$ could be obtained under a GP surrogate model by sampling functions from eqn. 13 and then optimizing them. In practice, eqn. 18 is difficult to compute, but there are several acquisition functions based on instantiations of this idea [86].

Illustrating BO acquisition and the exploration-exploitation tradeoff. Fig. 4a illustrates the EI acquisition function in a toy one-dimensional NPM space under a GP surrogate model with n=5



(a) iteration n

(b) iteration $n + 1$

Figure 4: Illustration of one iteration of BO where (i) the acquisition function is used to select the next NPM whilst balancing exploitation and exploration and (ii) the surrogate GP model based on the RBF kernel is updated to account for the newly acquired observation. (a) Iteration n. (b) Iteration $n + 1$, after the surrogate model is updated by the new observation (x_{n+1}, y_{n+1}) acquired to maximize the expected improvement (EI). In both (a) and (b), the top panel shows the surrogate model, and the bottom panel shows the expected improvement (EI) acquisition function. For comparison, in (a) we illustrate the NPM x_{n+1} that would have been selected under a pure exploitation or pure exploration acquisition strategy.

observations. The EI acquisition function exhibits two maxima. The first (global) maximum is in a region of the feature space where the predicted property $\hat{y}(x)$ is the largest. The second (local) maximum is where the uncertainty $\sigma(x)$ is largest. We select the NPM for the next experiment, x_{n+1}, as the one that maximizes the EI acquisition function. Fig. 4a shows the acquired NPM assuming the database of NPMs $\mathcal{X}$ covers all points on the domain shown. To illustrate how the EI acquisition strategy balances exploration and exploitation, for comparison, we also show the acquired NPM x_{n+1} if the acquisition strategy were purely exploration and purely exploitation. Pure exploration dictates $x_{n+1} = \arg\max \sigma(x)$, but this NPM has a poor property. Pure exploitation dictates $x_{n+1} = \arg\max \hat{y}(x)$, but this NPM is too close to an existing observation. EI balances the trade-off by picking an NPM with both a high uncertainty and high predicted property.

6 Experiments and Results

We now demonstrate Bayesian optimization by using it to efficiently search for covalent organic frameworks (COFs) for vehicular natural gas storage [2]. Our experiments below use the open data from Mercado et al. [28] and can be fully reproduced using our computer code at `github.com/aryandeshwal/BO_of_COFs`.

6.1 Experimental problem setup

Our goal is to efficiently search a database of COFs for the one with the largest methane deliverable capacity.

The database of COFs, $\mathcal{X}$. The database of COFs contains 69 840 2D and 3D predicted COF structures [28].

The methane deliverable capacity, y. The COF property we wish to maximize is the simulated deliverable capacity of methane [L STP CH_4/L COF] at 298 K under a 65 bar to 5.8 bar pressure swing. The deliverable capacity of the COF primarily determines the driving range of a vehicle on a "full" adsorbed natural gas fuel tank packed with the COF [2].

The COF vector representation, $\mathbf{x}$. We represent each COF structure with a vector $\mathbf{x} \in \mathbb{R}^{12}$ of structural and chemical features listed in Tab. 1 and computed by Mercado et al. [28]. N.b., we normalized each feature to lie in $[0, 1]$. This defines COF feature space.

The expensive objective function, $f(\mathbf{x})$. Evaluating the objective function f to give $y = f(\mathbf{x})$ involves conducting two grand-canonical Monte Carlo simulations of methane adsorption in the COF structure represented by **x**– one at (65 bar, 298 K) and one at (5.8 bar, 298 K). The deliverable capacity y then follows from the difference in simulated methane adsorption at the two conditions. The function f is expensive to evaluate, as the run time of these molecular simulations is on the order of hours.

Goal: data-efficient search for the optimal COF, $\mathbf{x}^*$. In an *exhaustive search*, we would conduct expensive molecular simulations to predict the methane deliverable capacity of each COF in the database—i.e., collect $\{(\mathbf{x}, y = f(\mathbf{x})) : \mathbf{x} \in \mathcal{X}\}$—to find the COF $\mathbf{x}^* \in \mathcal{X}$ with the highest deliverable capacity. In contrast to an exhaustive search, instead, our goal is to find the optimal COF $\mathbf{x}^*$ *efficiently*– while conducting expensive molecular simulations *in only a small proportion of the COFs*.

We hypothesize that BO will provide a simulation-efficient search for the optimal COF, $\mathbf{x}^*$. In reality, Mercado et. al. [28] already simulated methane adsorption in all of the COFs to compute their deliverable capacity. Thus, (i) we know the optimal COF $\mathbf{x}^*$ and (ii) instead of actually conducting molecular simulations of methane adsorption in a selected COF during the active search, we instead look up the result of the simulations (the deliverable capacity) from the data of Mercado et al. [28]. Each data lookup (equivalent to conducting simulations ourselves) represents conducting the two expensive molecular simulations of methane adsorption in a COF.In the general experimental setting where this data lookup is not possible, we will be performing molecular simulations for each selected COF. Note that, there is no impact on the BO solution as it only requires the property values of the selected COFs (and the how they are obtained is irrelevant). The exhaustive search of Mercado et al. [28] allows us to readily evaluate the simulation-efficiency of different search strategies to find the optimal COF(s). We will compare the search efficiency of BO to random search, an evolutionary algorithm, and one-shot supervised learning.

Table 1: Features comprising the vector representation of a COF, **x**, broken into those that capture its structure and chemical composition.

structural (geometrical)	chemical composition
void fraction	density carbon
density	density flourine
largest included sphere diameter	density hydrogen
largest free sphere diameter	density nitrogen
gravimetric surface area	density oxygen
	density sulfur
	density silicon

6.2 Search strategies

We use several different strategies to search for the optimal COF $\mathbf{x}^*$ exhibiting the highest methane deliverable capacity y in the database $\mathcal{X}$.

Random search. Random search is a naive baseline. At each iteration, we uniform randomly select an unevaluated COF from $\mathcal{X}$ to evaluate. Random search completely ignores the past observations in selecting a COF for the next evaluation, and thus is expected to perform poorly.

Bayesian optimization (BO). For BO, we employ (1) a Gaussian process (GP) with the Matérn kernel ($\nu = 2.5$) [80] as our surrogate model and (2) the expected improvement (EI) in eqn. 16 as our acquisition function. At each iteration of BO, we choose the hyperparameters of the Matern kernel (length-scale and signal variance; $\nu = 2.5$ fixed) by maximizing the marginal likelihood of the data under the GP. We implemented our BO procedure in the popular BoTorch library [87].

To initialize the GP for BO, we first randomly select ten COFs from the database and evaluate their methane deliverable capacity. The GP surrogate model is then trained on $\{(\mathbf{x}_i, y_i)\}_{i=1}^{10}$, which are included in evaluating the efficacy of BO based search strategy.

Evolutionary Search. We use Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [88] which is a state-of-the-art evolutionary search algorithm for rugged, non-convex objective functions. CMA-ES is an instantiation of the general evolutionary search framework where new inputs are explored in the search space by sampling from a Gaussian distribution that is updated over the search process. The key idea behind CMA-ES [88] is to update the mean of this distribution by maximizing the likelihood of the best observed inputs (i.e., NPM and their evaluations) and adapt the covariance matrix by maximizing the likelihood of the best observed search steps (i.e., result in uncovering NPMs with higher property values). CMA-ES has two hyper-parameters: initial sigma (standard deviation for each feature of the NPM representation, also known as step-size) and population size (number of candidate NPMs generated in each search iteration). We experimented with multiple values of these parameters by performing grid search and show results with the best found configuration of sigma = 0.2 and population size = 20.

One-shot supervised learning. While one-shot supervised learning does not constitute active search like BO, it is the most popular method for circumventing an exhaustive search for the optimal NPM using the high-fidelity evaluation method. We compare one-shot supervised learning with active search by varying the size of the training set (100, 200, 300, 400, and 500 COFs). Each COF selected to be a member of the training set counts in our search efficiency metrics because the molecular simulations must be conducted in it to predict its deliverable capacity. We use random forests (RFs) (100 trees, default parameters in `scikit-learn`) to build regression models to approximate $f(\mathbf{x})$ using the (differently sized) training sets. We then use the trained RF to estimate the deliverable capacities of the remaining, unevaluated COFs. Finally, we select and evaluate the unevaluated COF with the highest predicted (by the trained RF) deliverable capacity. This resembles a fully exploitative acquisition strategy. Thus, in our comparison of one-shot supervised learning with BO, we give each a fixed budget of n high-fidelity COF evaluations (for the case of one-shot supervised learning, n is the number of training data minus one) on which to base a decision of which unevalauted COF to acquire next.

For one-shot supervised learning, we use two different training set selection strategies: selection of NPMs from $\mathcal{X}$ to label (1) at uniform random and (2) according to a maximum-minimum diversity selection strategy [39, 89].

6.3 Results

We now execute each strategy to search for the optimal COF $\mathbf{x}^*$ exhibiting the highest methane deliverable capacity y in the database $\mathcal{X}$.

6.3.1 Search efficiency

The blue curves in Fig. 5 show the search efficiency of BO. Three different search performance metrics are shown as the number of COFs evaluated, n, ($=$ the number of BO iterations $=$ the number of simulations/"experiments") increases. The first metric in (a) is the maximum deliverable capacity among the acquired set of COFs, $\mathcal{X}_n$. The second and third metric show, among the entire data set $\mathcal{X}$, (b) the highest deliverable capacity ranking and (c) the fraction of the 100 COFs with the highest deliverable capacity, among $\mathcal{X}_n$. On average, BO acquires the optimal COF $\mathbf{x}^*$ and 25% of the top 100 ranked COFs after only 110 COF evaluations. After 500 COF evaluations, BO acquires 50% of the top 100 COFs in the data set. This illustrates how BO can provide a simulation-efficient search for the optimal COF as opposed to conducting an exhaustive search; BO picked out the top COF in the database of $\sim$70 000 COFs while evaluating less than 200!

BO also outperforms random search, evolutionary search, and the one-shot supervised learning approach. With the same number of COF evaluations (500), a random search (on average) acquires only the 170th ranked COF and 0.8% of the top 100 COFs. The performance of random search is poor because it ignores the past observations when it selects the next COF to evaluate. Evolutionary search and one-shot supervised learning (diverse training set) have a much better search performance than random search, acquiring the 14th/29th ranked COF and the top 15%/11% of the top 100 COFs on a budget of 500 evaluations. Though, evolutionary search nor the one-shot supervised learning strategies recover the optimal COF $\mathbf{x}^*$ after 500 evaluations. Thus, BO outperforms the baseline search methods of evolutionary search and one-shot supervised learning using both metrics of (a) the highest deliverable capacity in the acquired set and (b) the fraction of the top 100 COFs in the acquired set given a budget of 500 evaluations. N.b., BO is designed to optimize the former performance metric, but BO could potentially be improved to optimize a top-k metric [90]. With *very* few COF evaluations, the RF method with a diverse training set is more effective than BO, but BO outperforms the RF later with more evaluations.

We remark on the two different training set selection strategies for the one-shot supervised learning approach. The diverse selection of training data gives better search performance than the random selection of training data since it provides better coverage of the feature space for a finite budget of training data.

6.3.2 Visualizing the BO acquisition set

To understand the behavior of BO for searching the space of COFs for the optimal COF $\mathbf{x}^*$, we visualize the acquisition set of COFs in feature space as BO progresses.

First, we visualize the structure-property relationship $f(\mathbf{x})$ by approximating feature space using the first two principal components of the data $[\mathbf{x}_1\mathbf{x}_2\cdots\mathbf{x}_{|\mathcal{X}|}]$. Fig. 6a shows a 2D heat map depiction of $f(\mathbf{x})$ over the reduced 2D feature space. The color of each voxel in the reduced COF space indicates the average deliverable capacity of COFs that fall in that voxel of COF space.

Fig. 6b shows the acquired set of COFs, colored by deliverable capacity, at 20, 40, 60, and 80 iterations of BO. For reference, the gray background shows the coverage of COF space by all COFs in the dataset, shown in Fig. 6a. As BO progresses, it avoids selecting COFs from regions of COF space containing COFs that tend to exhibit low deliverable capacities, but well-samples regions containing COFs with high deliverable capacity. After 40 iterations, BO acquires COFs in low-performing regions of the feature space as exploration dominates exploitation of the region enriched with high-performing NPMs.

6.4 Conclusions from experiments

BO is an active search method to find the optimal NPM in a data set whilst evaluating, with some expensive method such as a molecular simulation, only a small fraction of the NPMs in the data set. BO achieves this by making acquisition decisions that take into account all past observations and balance exploration and exploitation. The adoption of BO could dramatically impact high-throughput computational screenings of NPMs by reducing the computing cost of finding the optimal NPM, allowing us to screen larger databases of NPMs, and enabling the use of higher-fidelity but more expensive molecular models and simulation methods. Notably, BO applies to NPM search in the experimental domain as well.

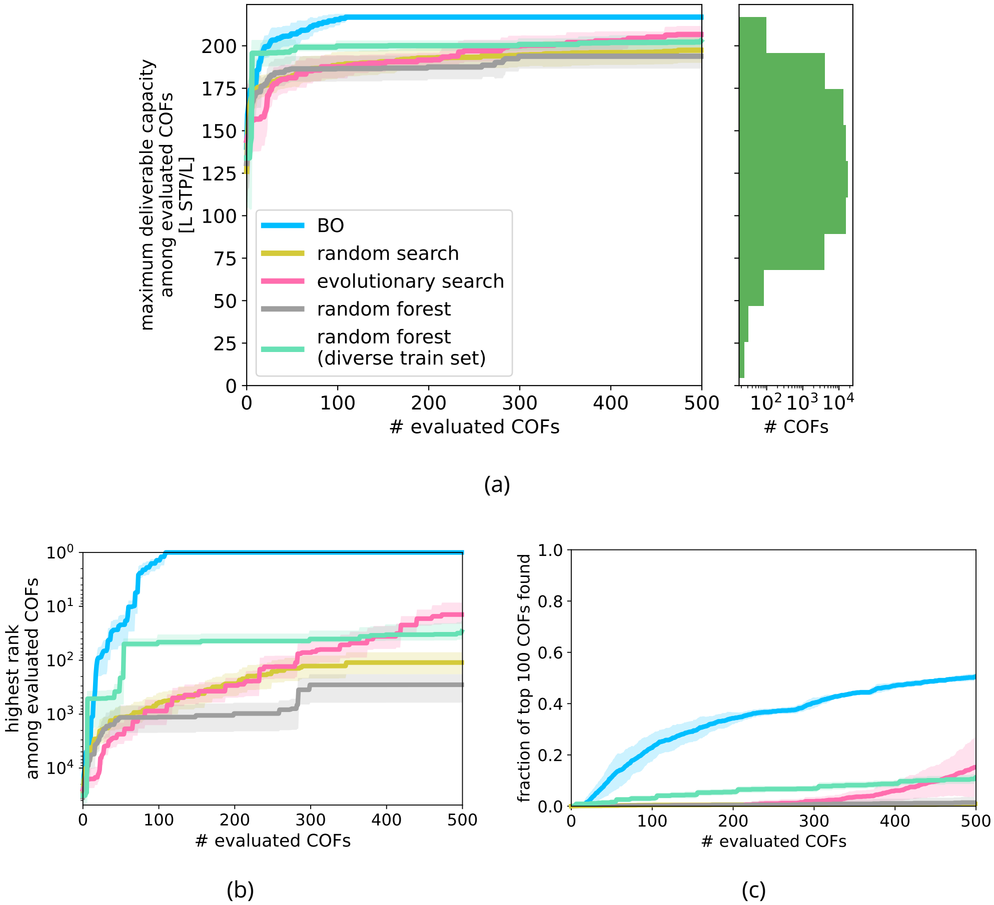
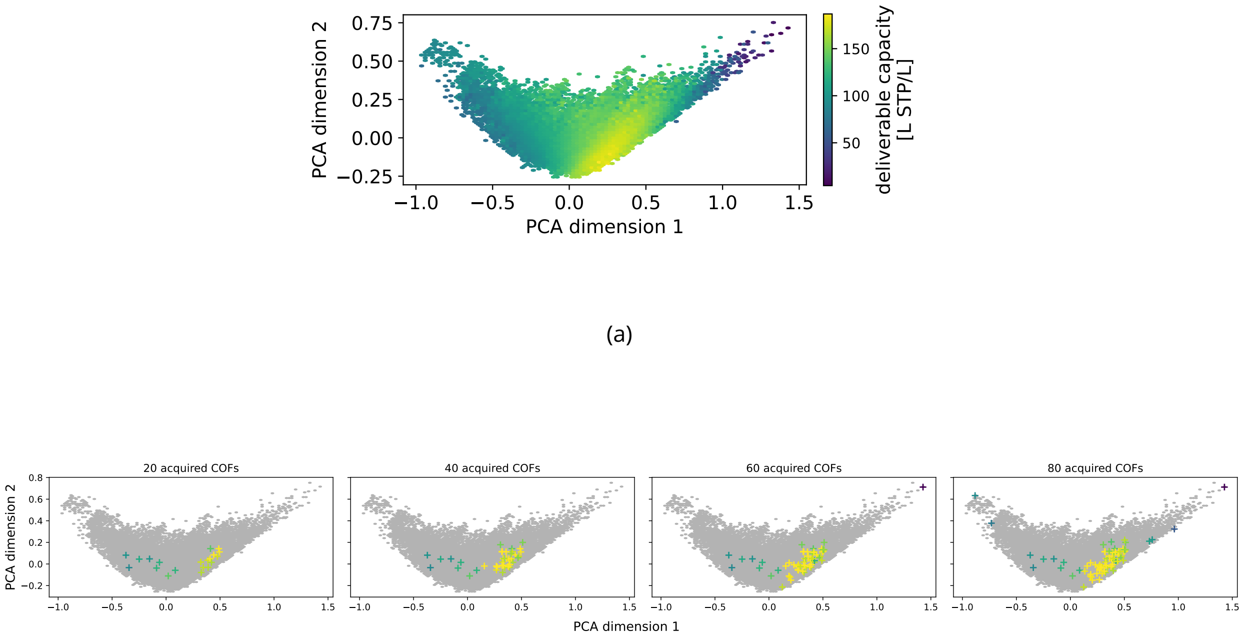


Figure 5: The search efficiency of BO in comparison with random search, evolutionary search, and one-shot supervised learning. The curves show search efficiency in terms of (a) the maximum deliverable capacity, (b) the highest ranking (among the entire data set $\mathcal{X}$) of the COF deliverable capacity, and (c) the fraction of the top 100 ranked (among $\mathcal{X}$) COFs– among the acquired COFs, in terms of the number of acquired/evaluated COFs. The shaded region shows the variance over 10 [stochastic] runs. To give (a) context, we show the distribution of the deliverable capacities among the COFs in the entire data set, $\mathcal{X}$, on the right.



(a)

(b)

Figure 6: Illustration of the COF acquisition set under BO. Each panel is a visualization of the first two principal components of COF feature space. (a) A heat map where the color of each hexagonal voxel indicates the average deliverable capacity of a COF that falls in that voxel. This is an attempt to visualize $f(\mathbf{x})$. (b) The points represent the acquired COFs at different stages of BO– after 20, 40, 60, and 80 iterations– and are colored according to the deliverable capacity (colorbar in (a) pertains). The gray background shows the region of the feature space covered by the COFs in (a).

7 Outlook

We explained the key ideas behind Bayesian optimization (BO) and advocated for its use to efficiently search databases of NPMs for the one with the optimal property, whilst synthesizing and evaluating the fewest NPMs. The ideas of BO, to sequentially, actively make intelligent decisions on which NPM to synthesize and evaluate based on the past experiments, can be applied to both the laboratory (driven by humans or robots [91–94]) and computational settings. The two core ingredients of BO are (1) a surrogate model that approximates the structure-property relationship and describes our uncertainty in it and (2) an acquisition function that scores the utility of evaluating each NPM next, designed to balance exploration and exploitation. We demonstrated BO of NPMs by using BO to search through a database of ca. 70 000 COFs to find the COF with the highest methane deliverable capacity. While preparing our article, Donval and Hand *et al*. [95] also demonstrated BO of MOFs and COFs for the acceleration of virtual screenings.

There are several extensions to and modifications of Bayesian optimization that are useful for different problem settings in NPM discovery:

- **Batch BO.** In standard BO, we select a single NPM to evaluate in each iteration. However, we may have parallel experimental resources to leverage to further accelerate the search for the optimal NPM. In *batch BO* [96–100], we can select multiple NPMs to synthesize and evaluate in parallel during each BO iteration.

- **Multi-fidelity BO.** Often, we have a choice of multiple experimental methods to evaluate the property of an NPM. These methods usually involve a tradeoff in resource cost and the accuracy of the evaluation. For example, a molecular simulation of gas adsorption in an NPM is a low-fidelity experiment (cheap, but inaccurate) while measurement of gas adsorption in an NPM in a physical laboratory is a high-fidelity experiment (costly, but accurate). Intuitively, it is possible to leverage low-fidelity experiments to prune NPMs with low property values and to identify promising NPM candidates that can be searched further using high-fidelity experiments. In *multi-fidelity BO* [101–110], we select both an NPM to evaluate and the fidelity of the experiment in each iteration. This allows optimization of the overall resource cost of experiments—of both low- and high-fidelity—for identifying high-performing NPMs.

- **Multi-objective BO.** We often need to optimize NPMs for multiple property objectives which are conflicting in nature and cannot be optimized simultaneously. For example, for gas separations, we often wish an NPM to have both a high selectivity and a high working capacity for the gas we wish to capture [51]. For multi-objective optimization problems, we need to find the Pareto optimal set of solutions.A solution is called Pareto optimal if it cannot be improved in any of the objectives without compromising some other objective. The goal of *multi-objective BO* is to find the optimal Pareto set of NPMs by minimizing the number of NPM evaluations [111–115]. Similarly, the ϵ-PAL algorithm [116] has recently been used to find the Pareto optimum polymers.

- **Constrained BO.** Possibly, some NPMs in the search space cannot be synthesized. More, often we cannot know if an NPM is synthesizable until we attempt its synthesis, which still incurs a cost. In this context, synthesizability is a black-box constraint over the search space.

In *constrained BO* [117–121], we perform BO where the synthesizability of an NPM cannot be verified without performing an experiment. The typical approach involves learned a statistical model based on the past evaluations of constraint(s) and selecting high-utility NPMs from the predicted feasible region (minimal to no constraint violation).

- **Cost-aware BO.** The evaluation cost can vary from one NPM to another (e.g., cost of synthesizing NPM). We would like to take this cost into account to reduce the costs incurred during the search for the optimal NPM. In *cost-aware BO* [122, 123], the acquisition strategy considers not only the information gain of acquiring an NPM but also the cost incurred to synthesize it and measure its property.

Some of the most popular software packages for BO include BoTorch [124], BayesOpt [125], and SMAC [126]. COMBO [127] is a BO library tailored to materials science.

In addition to efficiently searching for NPMs with optimal properties, BO is applicable to a wide variety of optimization problems in the chemical science. It has been used to find optimal reaction conditions [128, 129], the composition and processing conditions of materials to optimize their properties [93, 130], crystal structure prediction [131, 132], and calibrating density functional models [133]. For more general overview, see the reviews of Coley [134], Tsuda and coworkers [135], Frazier and Wang [136], and Lookman *et al*. [137].

The effectiveness of BO is predicted upon an accurate surrogate model of the structure-property relationship. In turn, the accuracy of the surrogate model is predicated on (i) an information-rich representation $\mathbf{x}$ of the NPM that encodes the salient features of its structure and chemical composition and (ii) a statistical model that (a) is sufficiently flexible/expressive to approximate the underlying objective function and (b) learns in a data-efficient manner. This gives important and currently active directions for future research. Particularly, engineering useful vector representations $\mathbf{x}$ of NPMs, using domain knowledge, is a very active research area [138, 139]. The representation should be invariant to rotations, translations, replications (if a crystal), and permutations of the list of atoms comprising the structure. Moreover, the mapping from NPM structures to feature vectors should be injective. Graph neural networks are becoming popular for instead *learning* vector representations of NPMs from their crystal structures represented as graphs with node labels. If there is variability in experiment and process conditions, we need to consider finding robust solutions through BO [140].

8 Acknowledgements

A.D. and J.D acknowledge support from NSF grants IIS-1845922, OAC-1910213. C.M.S. acknowledges support from NSF grant 1920945.

References

[1] Shengqian Ma and Hong-Cai Zhou. Gas storage in porous metal–organic frameworks for clean energy applications. *Chem. Commun.*, 46(1):44–53, 2010.

[2] Alexander Schoedel, Zhe Ji, and Omar M. Yaghi. The role of metal–organic frameworks in a carbon-neutral energy cycle. *Nature Energy*, 1(4), April 2016.

[3] Bin Li, Hui-Min Wen, Wei Zhou, and Banglin Chen. Porous metal–organic frameworks for gas storage and separation: What, how, and why? *The Journal of Physical Chemistry Letters*, 5(20):3468–3479, September 2014.

[4] Lauren E. Kreno, Kirsty Leong, Omar K. Farha, Mark Allendorf, Richard P. Van Duyne, and Joseph T. Hupp. Metal–organic framework materials as chemical sensors. *Chemical Reviews*, 112(2):1105–1125, November 2011.

[5] Leslie J. Murray, Mircea Dincă, and Jeffrey R. Long. Hydrogen storage in metal–organic frameworks. *Chemical Society Reviews*, 38(5):1294, 2009.

[6] Kenji Sumida, David L Rogow, Jarad A Mason, Thomas M McDonald, Eric D Bloch, Zoey R Herm, Tae-Hyun Bae, and Jeffrey R Long. Carbon dioxide capture in metal–organic frameworks. *Chemical Reviews*, 112(2):724–781, 2012.

[7] Fei-Yan Yi, Dongxiao Chen, Meng-Ke Wu, Lei Han, and Hai-Long Jiang. Chemical sensors based on metal–organic frameworks. *ChemPlusChem*, 81(8):675–690, 2016.

[8] Zhichao Hu, Benjamin J Deibert, and Jing Li. Luminescent metal–organic frameworks for chemical sensing and explosive detection. *Chemical Society Reviews*, 43(16):5815–5840, 2014.

[9] Christian S. Diercks and Omar M. Yaghi. The atom, the molecule, and the covalent organic framework. *Science*, 355(6328):eaal1585, March 2017.

[10] Xiao Feng, Xuesong Ding, and Donglin Jiang. Covalent organic frameworks. *Chemical Society Reviews*, 41(18):6010, 2012.

[11] Maria S. Lohse and Thomas Bein. Covalent organic frameworks: Structures, synthesis, and applications. *Advanced Functional Materials*, 28(33):1705553, July 2018.

[12] Shun Wan, Felipe Gándara, Atsushi Asano, Hiroyasu Furukawa, Akinori Saeki, Sanjeev K. Dey, Lei Liao, Michael W. Ambrogio, Youssry Y. Botros, Xiangfeng Duan, Shu Seki, J. Fraser Stoddart, and Omar M. Yaghi. Covalent organic frameworks with high charge carrier mobility. *Chemistry of Materials*, 23(18):4094–4097, September 2011.

[13] Daniele Ongari, Aliaksandr V. Yakutovich, Leopold Talirz, and Berend Smit. Building a consistent and reproducible database for adsorption evaluation in covalent–organic frameworks. *ACS Central Science*, 5(10):1663–1675, September 2019.

[14] Hiroyasu Furukawa, Kyle E. Cordova, Michael O'Keeffe, and Omar M. Yaghi. The chemistry and applications of metal-organic frameworks. *Science*, 341(6149):1230444, August 2013.

[15] David J. Tranchemontagne, Zheng Ni, Michael O'Keeffe, and Omar M. Yaghi. Reticular chemistry of metal–organic polyhedra. *Angewandte Chemie International Edition*, 47(28):5136–5147, June 2008.

[16] Tom Hasell and Andrew I Cooper. Porous organic cages: soluble, modular and molecular pores. *Nature Reviews Materials*, 1(9):1–14, 2016.

[17] Mian Li, Dan Li, Michael O'Keeffe, and Omar M. Yaghi. Topological analysis of metal–organic frameworks with polytopic linkers and/or multiple building units and the minimal transitivity principle. *Chemical Reviews*, 114(2):1343–1370, November 2013.

[18] Valentina Santolini, Marcin Miklitz, Enrico Berardo, and Kim E. Jelfs. Topological landscapes of porous organic cages. *Nanoscale*, 9(16):5280–5298, 2017.

[19] José L Segura, Sergio Royuela, and M Mar Ramos. Post-synthetic modification of covalent organic frameworks. *Chemical Society Reviews*, 48(14):3903–3945, 2019.

[20] Sukhendu Mandal, Srinivasan Natarajan, Prabu Mani, and Asha Pankajakshan. Post-synthetic modification of metal–organic frameworks toward applications. *Advanced Functional Materials*, 31(4):2006291, October 2020.

[21] Peter G. Boyd, Yongjin Lee, and Berend Smit. Computational development of the nanoporous materials genome. *Nature Reviews Materials*, 2(8), July 2017.

[22] Yamil J. Colón, Diego A. Gómez-Gualdrón, and Randall Q. Snurr. Topologically guided, automated construction of metal–organic frameworks and their evaluation for energy-related applications. *Crystal Growth & Design*, 17(11):5801–5810, September 2017.

[23] Lukas Turcani, Enrico Berardo, and Kim E. Jelfs. stk : A python toolkit for supramolecular assembly. *Journal of Computational Chemistry*, 39(23):1931–1942, September 2018.

[24] Peter G Boyd and Tom K Woo. A generalized method for constructing hypothetical nanoporous materials of any net topology from graph theory. *CrystEngComm*, 18(21):3777–3792, 2016.

[25] Richard Luis Martin and Maciej Haranczyk. Construction and characterization of structure models of crystalline porous polymers. *Crystal Growth & Design*, 14(5):2431–2440, April 2014.

[26] Arni Sturluson, Melanie T. Huynh, Alec R. Kaija, Caleb Laird, Sunghyun Yoon, Feier Hou, Zhenxing Feng, Christopher E. Wilmer, Yamil J. Colón, Yongchul G. Chung, Daniel W. Siderius, and Cory M. Simon. The role of molecular modelling and simulation in the discovery and deployment of metal-organic frameworks for gas storage and separation. *Molecular Simulation*, 45(14-15):1082–1121, August 2019.

[27] Hilal Daglar and Seda Keskin. Recent advances, opportunities, and challenges in high-throughput computational screening of MOFs for gas separations. *Coordination Chemistry Reviews*, 422:213470, November 2020.

[28] Rocío Mercado, Rueih-Sheng Fu, Aliaksandr V. Yakutovich, Leopold Talirz, Maciej Haranczyk, and Berend Smit. In silico design of 2d and 3d covalent organic frameworks for methane storage applications. *Chemistry of Materials*, 30(15):5069–5086, June 2018.

[29] Kevin Maik Jablonka, Daniele Ongari, Seyed Mohamad Moosavi, and Berend Smit. Big-data science in porous materials: Materials genomics and machine learning. *Chemical Reviews*, 120(16):8066–8129, June 2020.

[30] Robert Pollice, Gabriel dos Passos Gomes, Matteo Aldeghi, Riley J. Hickman, Mario Krenn, Cyrille Lavigne, Michael Lindner-D'Addario, AkshatKumar Nigam, Cher Tian Ser, Zhenpeng Yao, and Alán Aspuru-Guzik. Data-driven strategies for accelerated materials design. *Accounts of Chemical Research*, 54(4):849–860, February 2021.

[31] Sanggyu Chong, Sangwon Lee, Baekjun Kim, and Jihan Kim. Applications of machine learning in metal-organic frameworks. *Coordination Chemistry Reviews*, 423:213487, 2020.

[32] Siwar Chibani and François-Xavier Coudert. Machine learning approaches for the prediction of materials properties. *APL Materials*, 8(8):080701, August 2020.

[33] Seyed Mohamad Moosavi, Kevin Maik Jablonka, and Berend Smit. The role of machine learning in the understanding and design of materials. *Journal of the American Chemical Society*, 142(48):20273–20287, November 2020.

[34] Krishnendu Mukherjee and Yamil J. Colón. Machine learning and descriptor selection for the computational discovery of metal-organic frameworks. *Molecular Simulation*, pages 1–21, April 2021.

[35] Zenan Shi, Wenyuan Yang, Xiaomei Deng, Chengzhi Cai, Yaling Yan, Hong Liang, Zili Liu, and Zhiwei Qiao. Machine-learning-assisted high-throughput computational screening of high performance metal–organic frameworks. *Molecular Systems Design & Engineering*, 5(4):725–742, 2020.

[36] Marcel F Langer, Alex Goeßmann, and Matthias Rupp. Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning. *arXiv preprint arXiv:2003.12081*, 2020.

[37] Christopher R Collins, Geoffrey J Gordon, O Anatole Von Lilienfeld, and David J Yaron. Constant size descriptors for accurate machine learning models of molecular properties. *The Journal of Chemical Physics*, 148(24):241718, 2018.

[38] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

[39] Cory M. Simon, Rocio Mercado, Sondre K. Schnell, Berend Smit, and Maciej Haranczyk. What are the best materials to separate a xenon/krypton mixture? *Chemistry of Materials*, 27(12):4459–4475, June 2015.

[40] Seyed Mohamad Moosavi, Aditya Nandy, Kevin Maik Jablonka, Daniele Ongari, Jon Paul Janet, Peter G. Boyd, Yongjin Lee, Berend Smit, and Heather J. Kulik. Understanding the diversity of the metal-organic framework ecosystem. *Nature Communications*, 11(1), August 2020.

[41] Eun Hyun Cho, Xuepeng Deng, Changlong Zou, and Li-Chiang Lin. Machine learning-aided computational study of metal–organic frameworks for sour gas sweetening. *The Journal of Physical Chemistry C*, 2020.

[42] Jake Burner, Ludwig Schwiedrzik, Mykhaylo Krykunov, Jun Luo, Peter G. Boyd, and Tom K. Woo. High-performing deep learning regression models for predicting low-pressure CO2 adsorption

properties of metal–organic frameworks. *The Journal of Physical Chemistry C*, 124(51):27996–28005, December 2020.

[43] Benjamin J. Bucior, N. Scott Bobbitt, Timur Islamoglu, Subhadip Goswami, Arun Gopalan, Taner Yildirim, Omar K. Farha, Neda Bagheri, and Randall Q. Snurr. Energy-based descriptors to rapidly predict hydrogen storage in metal–organic frameworks. *Molecular Systems Design & Engineering*, 4(1):162–174, 2019.

[44] Ryther Anderson, Jacob Rodgers, Edwin Argueta, Achay Biong, and Diego A. Gómez-Gualdrón. Role of pore chemistry and topology in the CO2capture capabilities of MOFs: From molecular simulation to machine learning. *Chemistry of Materials*, 30(18):6325–6337, August 2018.

[45] George S. Fanourgakis, Konstantinos Gkagkas, Emmanuel Tylianakis, and George Froudakis. A generic machine learning algorithm for the prediction of gas adsorption in nanoporous materials. *The Journal of Physical Chemistry C*, 124(13):7117–7126, March 2020.

[46] Maryam Pardakhti, Ehsan Moharreri, David Wanik, Steven L. Suib, and Ranjan Srivastava. Machine learning using combined structural and chemical descriptors for prediction of methane adsorption performance of metal organic frameworks (MOFs). *ACS Combinatorial Science*, 19(10):640–645, September 2017.

[47] Michael Fernandez and Amanda S. Barnard. Geometrical properties can predict CO2 and n2 adsorption performance of metal–organic frameworks (MOFs) at low pressure. *ACS Combinatorial Science*, 18(5):243–252, April 2016.

[48] Hana Dureckova, Mykhaylo Krykunov, Mohammad Zein Aghaji, and Tom K. Woo. Robust machine learning models for predicting high CO2 working capacity and CO2/h2 selectivity of gas adsorption in metal organic frameworks for precombustion carbon capture. *The Journal of Physical Chemistry C*, 123(7):4133–4139, January 2019.

[49] Xiangyu Zhang, Jing Cui, Kexin Zhang, Jiasheng Wu, and Yongjin Lee. Machine learning prediction on properties of nanoporous materials utilizing pore geometry barcodes. *Journal of Chemical Information and Modeling*, 59(11):4636–4644, October 2019.

[50] Tu C. Le and David A. Winkler. Discovery and optimization of materials using evolutionary approaches. *Chemical Reviews*, 116(10):6107–6132, May 2016.

[51] Yongchul G. Chung, Diego A. Gómez-Gualdrón, Peng Li, Karson T. Leperi, Pravas Deria, Hongda Zhang, Nicolaas A. Vermeulen, J. Fraser Stoddart, Fengqi You, Joseph T. Hupp, Omar K. Farha, and Randall Q. Snurr. In silico discovery of metal-organic frameworks for precombustion CO2 capture using a genetic algorithm. *Science Advances*, 2(10):e1600909, October 2016.

[52] Sean P. Collins, Thomas D. Daff, Sarah S. Piotrkowski, and Tom K. Woo. Materials design by evolutionary optimization of functional groups in metal-organic frameworks. *Science Advances*, 2(11):e1600954, November 2016.

[53] Seyed Mohamad Moosavi, Arunraj Chidambaram, Leopold Talirz, Maciej Haranczyk, Kyriakos C. Stylianou, and Berend Smit. Capturing chemical intuition in synthesis of metal-organic frameworks. *Nature Communications*, 10(1), February 2019.

[54] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[55] Xiangyu Zhang, Kexin Zhang, and Yongjin Lee. Machine learning enabled tailor-made design of application-specific metal–organic frameworks. *ACS Applied Materials & Interfaces*, 12(1):734–743, December 2019.

[56] Xiangyu Zhang, Kexin Zhang, Hyeonsuk Yoo, and Yongjin Lee. Machine learning-driven discovery of metal–organic frameworks for efficient CO2 capture in humid condition. *ACS Sustainable Chemistry & Engineering*, February 2021.

[57] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[58] Eugene N. Muratov, Jürgen Bajorath, Robert P. Sheridan, Igor V. Tetko, Dmitry Filimonov, Vladimir Poroikov, Tudor I. Oprea, Igor I. Baskin, Alexandre Varnek, Adrian Roitberg, Olexandr Isayev, Stefano Curtalolo, Denis Fourches, Yoram Cohen, Alan Aspuru-Guzik, David A. Winkler, Dimitris Agrafiotis, Artem Cherkasov, and Alexander Tropsha. QSAR without borders. *Chemical Society Reviews*, 49(11):3525–3564, 2020.

[59] Tu Le, V. Chandana Epa, Frank R. Burden, and David A. Winkler. Quantitative structure-property relationship modeling of diverse materials properties. *Chemical Reviews*, 112(5):2889–2919, January 2012.

[60] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[61] Apoorv Agnihotri and Nipun Batra. Exploring bayesian optimization. *Distill*, 2020. https://distill.pub/2020/bayesian-optimization.

[62] Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87(18), May 2013.

[63] Lev Sarkisov, Rocio Bueno-Perez, Mythili Sutharson, and David Fairen-Jimenez. Materials informatics with PoreBlazer v4.0 and the CSD MOF database. *Chemistry of Materials*, 32(23):9849–9867, November 2020.

[64] Thomas F Willems, Chris H Rycroft, Michaeel Kazi, Juan C Meza, and Maciej Haranczyk. Algorithms and tools for high-throughput geometry-based analysis of crystalline porous materials. *Microporous and Mesoporous Materials*, 149(1):134–141, 2012.

[65] Lev Sarkisov and Alex Harrison. Computational structure characterisation tools in application to ordered and disordered porous materials. *Molecular Simulation*, 37(15):1248–1257, 2011.

[66] Andrew S. Rosen, Shaelyn M. Iyer, Debmalya Ray, Zhenpeng Yao, Alán Aspuru-Guzik, Laura Gagliardi, Justin M. Notestein, and Randall Q. Snurr. Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery. *Matter*, April 2021.

[67] Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical Review Letters*, 120(14), April 2018.

[68] Ali Raza, Faaiq Waqar, Arni Sturluson, Cory Simon, and Xiaoli Fern. Towards explainable message passing networks for predicting carbon dioxide adsorption in metal-organic frameworks. *arXiv preprint arXiv:2012.03723*, 2020.

[69] Eun Hyun Cho and Li-Chiang Lin. Nanoporous material recognition via 3d convolutional neural networks: Prediction of adsorption properties. *The Journal of Physical Chemistry Letters*, 12(9):2279–2285, 2021.

[70] Arni Sturluson, Melanie T. Huynh, Arthur H. P. York, and Cory M. Simon. Eigencages: Learning a latent space of porous cage molecules. *ACS Central Science*, 4(12):1663–1676, December 2018.

[71] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[72] Christopher R. Collins, Geoffrey J. Gordon, O. Anatole von Lilienfeld, and David J. Yaron. Constant size descriptors for accurate machine learning models of molecular properties. *The Journal of Chemical Physics*, 148(24):241718, June 2018.

[73] Zhenpeng Yao, Benjamín Sánchez-Lengeling, N. Scott Bobbitt, Benjamin J. Bucior, Sai Govind Hari Kumar, Sean P. Collins, Thomas Burns, Tom K. Woo, Omar K. Farha, Randall Q. Snurr, and Alán Aspuru-Guzik. Inverse design of nanoporous crystalline reticular materials with deep generative models. *Nature Machine Intelligence*, 3(1):76–86, January 2021.

[74] Michael Fernandez, Nicholas R. Trefiak, and Tom K. Woo. Atomic property weighted radial distribution functions descriptors of metal–organic frameworks for the prediction of gas uptake capacity. *The Journal of Physical Chemistry C*, 117(27):14095–14105, July 2013.

[75] Yongjin Lee, Senja D. Barthel, Paweł Dłotko, Seyed Mohamad Moosavi, Kathryn Hess, and Berend Smit. High-throughput screening approach for nanoporous materials genome using topological data analysis: Application to zeolites. *Journal of Chemical Theory and Computation*, 14(8):4427–4437, July 2018.

[76] Felix Faber, Alexander Lindmaa, O. Anatole von Lilienfeld, and Rickard Armiento. Crystal structure representations for machine learning models of formation energies. *International Journal of Quantum Chemistry*, 115(16):1094–1101, April 2015.

[77] Uc berkeley cs188 intro to ai – course materials.

[78] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1015–1022. Omnipress, 2010.

[79] Burr Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[80] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

[81] Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002.

[82] Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, and Bastian Rieck. Graph kernels: state-of-the-art and future challenges. *arXiv preprint arXiv:2011.03854*, 2020.

[83] Hiroshi Ohno and Yusuke Mukae. Machine learning approach for prediction and search: Application to methane storage in a metal–organic framework. *The Journal of Physical Chemistry C*, 120(42):23963–23968, October 2016.

[84] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[85] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV*, volume 6314 of *Lecture Notes in Computer Science*, pages 143–156. Springer, 2010.

[86] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *International Conference on Machine Learning*, pages 3627–3635. PMLR, 2017.

[87] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

[88] Nikolaus Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation*, pages 75–102, 2006.

[89] Daniel Cosmin Porumbel, Jin-Kao Hao, and Fred Glover. A simple and effective algorithm for the maxmin diversity problem. *Annals of Operations Research*, 186(1):275, 2011.

[90] Quoc Phong Nguyen, Sebastian Tay, Bryan Kian Hsiang Low, and Patrick Jaillet. Top-k ranking bayesian optimization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 9135–9143. AAAI Press, 2021.

[91] R. L. Greenaway, V. Santolini, M. J. Bennison, B. M. Alston, C. J. Pugh, M. A. Little, M. Miklitz, E. G. B. Eden-Rump, R. Clowes, A. Shakil, H. J. Cuthbertson, H. Armstrong, M. E. Briggs, K. E. Jelfs, and A. I. Cooper. High-throughput discovery of organic cages and catenanes using computational screening fused with robotic synthesis. *Nature Communications*, 9(1), July 2018.

[92] Connor W. Coley, Dale A. Thomas, Justin A. M. Lummiss, Jonathan N. Jaworski, Christopher P. Breen, Victor Schultz, Travis Hart, Joshua S. Fishman, Luke Rogers, Hanyu Gao, Robert W. Hicklin, Pieter P. Plehiers, Joshua Byington, John S. Piotti, William H. Green, A. John Hart, Timothy F.

Jamison, and Klavs F. Jensen. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453):eaax1566, August 2019.

[93] B. P. MacLeod, F. G. L. Parlane, T. D. Morrissey, F. Häse, L. M. Roch, K. E. Dettelbach, R. Moreira, L. P. E. Yunker, M. B. Rooney, J. R. Deeth, V. Lai, G. J. Ng, H. Situ, R. H. Zhang, M. S. Elliott, T. H. Haley, D. J. Dvorak, A. Aspuru-Guzik, J. E. Hein, and C. P. Berlinguette. Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, 6(20):eaaz8867, May 2020.

[94] Rebecca L. Greenaway and Kim E. Jelfs. Integrating computational and experimental workflows for accelerated organic materials discovery. *Advanced Materials*, 33(11):2004831, February 2021.

[95] Gael Donval, Calum Hand, James Hook, Emiko Dupont, Malena Sabate Landman, Malina Freitag, Matthew Lennox, and Tina Düren. Autonomous exploration and identification of high performing adsorbents using active learning. 2021.

[96] Javad Azimi, Alan Fern, and Xiaoli Z. Fern. Batch bayesian optimization via simulation matching. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 109–117. Curran Associates, Inc., 2010.

[97] Tarun Kathuria, Amit Deshpande, and Pushmeet Kohli. Batched gaussian process bandit optimization via determinantal point processes. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4206–4214, 2016.

[98] Christof Angermüller, David Belanger, Andreea Gane, Zelda Mariet, David Dohan, Kevin Murphy, Lucy Colwell, and D. Sculley. Population-based black-box optimization for biological sequence design. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 324–334. PMLR, 2020.

[99] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Mercer features for efficient combinatorial bayesian optimization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 7210–7218. AAAI Press, 2021.

[100] Florian Häse, Loïc M. Roch, Christoph Kreisbeck, and Alán Aspuru-Guzik. Phoenics: A bayesian optimizer for chemistry. *ACS Central Science*, 4(9):1134–1145, 2018.

[101] Rémi Lam, Douglas L Allaire, and et al. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015.

[102] Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, and et al. Gaussian process bandit optimisation with multi-fidelity evaluations. In *Conference on Neural Information Processing Systems*, 2016.

[103] Yehong Zhang, Trong Nghia Hoang, and et al. Information-based multi-fidelity Bayesian optimization. In *Conference on Neural Information Processing Systems Workshop on Bayesian Optimization*, 2017.

[104] Jialin Song, Yuxin Chen, and Yisong Yue. A general framework for multi-fidelity Bayesian optimization with Gaussian processes. *International Conference on Artificial Intelligence and Statistics*, 2019.

[105] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity Bayesian optimization with max-value entropy search. *arXiv:1901.08275*, 2019.

[106] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Multi-fidelity multi-objective Bayesian optimization: An output space entropy search approach. In *AAAI conference on Artificial Intelligence (AAAI)*, 2020.

[107] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Information-theoretic multi-objective Bayesian optimization with continuous approximations. *CoRR*, abs/2009.05700, 2020.

[108] H. C. Herbol, P. Clancy, and M. Poloczek. Cost-effective materials discovery: Bayesian optimization across information sources. *Mater. Horiz.*, 7:2113–2123, 2020.

[109] Olga Egorova, Roohollah Hafizi, David C. Woods, and Graeme M. Day. Multifidelity statistical machine learning for molecular crystal structure prediction. *The Journal of Physical Chemistry A*, 124(39):8065–8078, 2020. PMID: 32881496.

[110] Anh Tran, Julien Tranchida, Tim Wildey, and Aidan P Thompson. Multi-fidelity machine-learning with uncertainty quantification and bayesian optimization for materials design: Application to ternary random alloys. *The Journal of Chemical Physics*, 153(7):074705, 2020.

[111] Daniel Hernández-Lobato, Jose Hernandez-Lobato, Amar Shah, and Ryan Adams. Predictive entropy search for multi-objective Bayesian optimization. In *ICML*, pages 1492–1501, 2016.

[112] Belakaria et al. Max-value entropy search for multi-objective Bayesian optimization. In *NeurIPS*, 2019.

[113] Syrine Belakaria, Aryan Deshwal, Nitthilan Kannappan Jayakodi, and Janardhan Rao Doppa. Uncertainty-aware search framework for multi-objective Bayesian optimization. In *AAAI*, 2020.

[114] Artur M. Schweidtmann, Adam D. Clayton, Nicholas Holmes, Eric Bradford, Richard A. Bourne, and Alexei A. Lapkin. Machine learning meets continuous flow chemistry: Automated optimization towards the pareto front of multiple objectives. *Chemical Engineering Journal*, 352:277–282, 2018.

[115] Florian Häse, Loïc M Roch, and Alán Aspuru-Guzik. Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories. *Chemical science*, 9(39):7642–7655, 2018.

[116] Kevin Maik Jablonka, Giriprasad Melpatti Jothiappan, Shefang Wang, Berend Smit, and Brian Yoo. Bias free multiobjective active learning for materials design and discovery. *Nature communications*, 12(1):1–10, 2021.

[117] Valerio Perrone, Iaroslav Shcherbatyi, Rodolphe Jenatton, Cédric Archambeau, and Matthias W. Seeger. Constrained bayesian optimization with max-value entropy search. *CoRR*, abs/1910.07003, 2019.

[118] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective Bayesian optimization with constraints. *CoRR*, abs/2009.01721, 2020.

[119] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Uncertainty aware search framework for multi-objective Bayesian optimization with constraints. *CoRR*, abs/2008.07029, 2020.

[120] Zhiyuan Zhou, Syrine Belakaria, Aryan Deshwal, Wookpyo Hong, Janardhan Rao Doppa, Partha Pratim Pande, and Deukhyoun Heo. Design of multi-output switched-capacitor voltage regulator via machine learning. In *DATE*, 2020.

[121] Syrine Belakaria, Derek Jackson, Yue Cao, Janardhan Rao Doppa, and Xiaonan Lu. Machine learning enabled fast multi-objective optimization for electrified aviation power system design. In *ECCE*, 2020.

[122] Eric Hans Lee, Valerio Perrone, Cédric Archambeau, and Matthias W. Seeger. Cost-aware bayesian optimization. *CoRR*, abs/2003.10870, 2020.

[123] Gauthier Guinet, Valerio Perrone, and Cédric Archambeau. Pareto-efficient acquisition functions for cost-aware bayesian optimization. *CoRR*, abs/2011.11456, 2020.

[124] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

[125] Ruben Martinez-Cantin. Bayesopt: a bayesian optimization library for nonlinear optimization, experimental design and bandits. *J. Mach. Learn. Res.*, 15(1):3735–3739, 2014.

[126] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. Technical Report TR-2010-10, University of British Columbia, Department of Computer Science, 2010.

[127] Tsuyoshi Ueno, Trevor David Rhone, Zhufeng Hou, Teruyasu Mizoguchi, and Koji Tsuda. COMBO: An efficient bayesian optimization library for materials science. *Materials Discovery*, 4:18–21, June 2016.

[128] Benjamin J. Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I. Martinez Alvarado, Jacob M. Janey, Ryan P. Adams, and Abigail G. Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, February 2021.

[129] Artur M. Schweidtmann, Adam D. Clayton, Nicholas Holmes, Eric Bradford, Richard A. Bourne, and Alexei A. Lapkin. Machine learning meets continuous flow chemistry: Automated optimiza-

tion towards the pareto front of multiple objectives. *Chemical Engineering Journal*, 352:277–282, November 2018.

[130] Benjamin Burger, Phillip M. Maffettone, Vladimir V. Gusev, Catherine M. Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M. Alston, Buyi Li, Rob Clowes, Nicola Rankin, Brandon Harris, Reiner Sebastian Sprick, and Andrew I. Cooper. A mobile robotic chemist. *Nature*, 583(7815):237–241, July 2020.

[131] Tomoki Yamashita, Nobuya Sato, Hiori Kino, Takashi Miyake, Koji Tsuda, and Tamio Oguchi. Crystal structure prediction accelerated by bayesian optimization. *Physical Review Materials*, 2(1), January 2018.

[132] Edward Pyzer-Knapp, Graeme Day, Linjiang Chen, and Andrew I Cooper. Distributed multi-objective bayesian optimization for the intelligent navigation of energy structure function maps for efficient property discovery. *ChemRxiv*, 2020.

[133] R. A. Vargas-Hernández. Bayesian optimization for calibrating and selecting hybrid-density functional models. *The Journal of Physical Chemistry A*, 124(20):4053–4061, April 2020.

[134] Connor W. Coley. Defining and exploring chemical spaces. *Trends in Chemistry*, 3(2):133–145, February 2021.

[135] Kei Terayama, Masato Sumita, Ryo Tamura, and Koji Tsuda. Black-box optimization for automated discovery. *Accounts of Chemical Research*, 54(6):1334–1346, 2021.

[136] Peter I Frazier and Jialei Wang. Bayesian optimization for materials design. In *Information Science for Materials Discovery and Design*, pages 45–75. Springer, 2016.

[137] Turab Lookman, Prasanna V Balachandran, Dezhen Xue, and Ruihao Yuan. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Computational Materials*, 5(1):1–17, 2019.

[138] Florian Häse, Loïc M Roch, and Alán Aspuru-Guzik. Gryffin: An algorithm for bayesian optimization for categorical variables informed by physical intuition with applications to chemistry. *arXiv preprint arXiv:2003.12127*, 2020.

[139] Aryan Deshwal, Syrine Belakaria, and Janardhan Rao Doppa. Bayesian optimization over hybrid spaces. In *Proceedings of International Conference on Machine Learning (ICML)*, 2021.

[140] Matteo Aldeghi, Florian Häse, Riley J Hickman, Isaac Tamblyn, and Alán Aspuru-Guzik. Golem: An algorithm for robust experiment and process optimization. *arXiv preprint arXiv:2103.03716*, 2021.