Olympus, enhanced: benchmarking mixed-parameter and multi-objective optimization in chemistry and materials science

Riley J. Hickman,[1, 2, 3, *] Priyansh Parakh,[1, 4] Austin Cheng,[1, 2, 3] Qianxiang Ai,[5] Joshua Schrier,[5] Matteo Aldeghi,[6] and Alán Aspuru-Guzik[1, 2, 3, 7, 8, 9, †]

[1] *Chemical Physics Theory Group, Department of Chemistry,*
University of Toronto, Toronto, ON M5S 3H6, Canada
[2] *Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4, Canada*
[3] *Vector Institute for Artificial Intelligence, Toronto, ON M5S 1M1, Canada*
[4] *Faculty of Applied Science and Engineering, University of Toronto, ON M5S 1A4, Canada*
[5] *Department of Chemistry, Fordham University, 441 E. Fordham Road, The Bronx, New York 10458, United States*
[6] *Google Research, Mountain View, CA 94043, United States*
[7] *Department of Chemical Engineering & Applied Chemistry,*
University of Toronto, Toronto, ON M5S 3E5, Canada
[8] *Department of Materials Science & Engineering,*
University of Toronto, Toronto, ON M5S 3E4, Canada
[9] *Lebovic Fellow, Canadian Institute for Advanced Research, Toronto, ON M5G 1Z8, Canada*

(Dated: May 17, 2023)

Experiment planning algorithms are a required component of autonomous platforms for scientific discovery. Selecting a suitable optimization algorithm for a novel application is an important yet difficult choice a researcher has to make based on past empirical performance on similar tasks. To facilitate the evaluation of various algorithms on chemistry and materials science optimization tasks, we previously introduced Olympus (*Mach. Learn.: Sci. Technol.* **2**, 035021, 2021), a Python package providing a consistent and easy-to-use interface to numerous optimization algorithms and benchmark datasets. While the original package was limited to continuous parameters and single objectives, in this work we expand Olympus' capabilities to include mixed (continuous, discrete, and categorical) parameter types and multiple objectives. Several experiment planning algorithms already contained in Olympus are extended to handle categorical and discrete parameter types, and five additional planners are implemented (23 in total). We also provide 23 additional benchmark datasets taken from the chemistry and materials science literature (33 in total), covering a wide range of research areas, from chemical reaction optimization to materials manufacturing. Finally, the visualization capabilities of Olympus are enhanced to allow for easy inspection of the results, and the core functionality of the package is embedded in a Streamlit web application for code-free usage. We demonstrate how Olympus enables researchers to rapidly benchmark different optimization strategies and gain insight into their behavior by focusing on two case studies: the optimization of a Suzuki-Miyaura cross-coupling reaction with categorical reaction conditions, and the multi-objective optimization of redox-active materials. The updated Olympus package provides practitioners with a large suite of tools to efficiently benchmark and analyze experiment planning algorithms on mixed-parameter and multi-objective optimization tasks.

I. INTRODUCTION

Many challenges encountered in chemistry and materials science can be framed as optimization tasks. In fact, due to its ubiquity in all quantitative disciplines, optimization is a fundamental area of applied numerical science. Solving an optimization problem involves the systematic selection of input parameters from an allowed set, with the goal of identifying the parameters that produce the most desirable outcome. The property being optimized is typically referred to as the *objective function*, or *figure of merit*. Advances in the field of optimization research have consistently enabled technologies with significant societal impact. For instance, stochastic gradient descent is the key optimization algorithm that enables the training of deep neural networks.[1–3] Non-linear optimization strategies are also used heavily in molecular modelling, which is indispensable to the fields of computational chemistry, drug design, materials science, and computational biology.[4]

A subset of optimization problems of particular interest in the experimental sciences are global, black-box problems, which involve objective functions whose structure is *a priori* unknown, and must be resolved by sequential (oftentimes noisy) measurement. Prevalent examples include chemical reaction optimization (the maximization of the yield of the desired chemical product given a set of allowed conditions),[5–16] optimization of properties of advanced materials or functional molecules, such as photovoltaics,[17–19] nanomaterials,[20–24] and catalysts,[25,26] among others.[27–32] Model-based optimization approaches, such as Bayesian optimization, are proving particularly suited to black-box optimization tasks encountered in experimental sciences. Recently, the com-

[*] riley.hickman@mail.utoronto.ca
[†] alan@aspuru.com

bination of model-based optimization with automated laboratory equipment has shown potential for the development of fully autonomous research platforms, also referred to as self-driving laboratories or materials acceleration platforms.[33–39]

Although the mentioned experimental science applications are diverse in discipline, they have several characteristics in common. In particular, they typically (i) are black-box problems with no analytical gradient information available, (ii) involve expensive-to-evaluate objectives (with respect to monetary cost, time, or other resources), (iii) consist of several input parameter types (continuous, discrete, and categorical), (iv) involve measurement procedures which are subject to noise, and (v) involve multiple, potentially competing objectives that must be optimized simultaneously.

While some optimization strategies can perform well on a wide range of problems, selection of the best strategy for a novel problem remains non-trivial.[40,41] In part, this is due to the structure of the objective function being *a priori* unknown, which renders comparison to related problems challenging. As such, the selection of strategies for a novel problem currently relies heavily on user preference, the availability of user-friendly optimization software, and heuristics, as opposed to empirical evidence. In addition, benchmarking optimization strategies on real-world experimental problems is often prohibitively expensive as a result of the cost associated with experimental measurements. Thus, benchmarks continue to be carried out using analytical functions. While these benchmarks are convenient, given that their values can be computed instantaneously, they often do not constitute realistic optimization tasks encountered in real-world applications.

In the spirit of providing researchers with the ability to benchmark optimization strategies on tasks derived from real-world data, our group recently introduced Olympus.[42,43] This Python package provides users with the ability to benchmark the performance of optimization algorithms on emulated experiments from chemistry and materials science (core classes and their interactions are shown in Figure 1). Yet, many features of practical relevance are still lacking from Olympus, preventing researchers from fully capitalizing on its intended usage. In this work, we detail several recent extensions to Olympus. In summary, we contribute the following:

- Access to additional parameter types, including discrete, categorical, and ordinal parameters.

- 5 additional Bayesian optimization algorithms are implemented, and existing algorithms are updated for compatibility with the above-mentioned parameter types.

- 23 additional real-world datasets curated from the chemistry and materials science literature, and 10 additional analytical benchmark surfaces.

- Compatibility with multi-objective or Pareto optimization via an achievement scalarizing function (ASF) module.

- Expanded functionality for plotting and analyzing the results of optimization benchmarks, including tools for visualizing the result of experiments with mixed-parameters and multiple objectives.

- A Streamlit webapp providing code-free access to Olympus.

This manuscript is organized as follows. In Section II we review related efforts in benchmarking computational tools in the natural sciences. Section III delineates the updates to the Olympus package. In Section IV, we examine two case studies, which showcase the ease with which users can benchmark the performance of experiment planning strategies using Olympus. First, we compare the performance of several algorithms that support categorical parameters on the optimization of a Pd-catalyzed Suzuki-Miyaura cross-coupling reaction. Second, we show how the choice of ASF affects the multi-objective design of redox-active materials for non-aqueous flow batteries.

II. RELATED WORK

Research in natural science disciplines has routinely produced efforts to benchmark computational methods. The area of computational chemistry has relied for decades on thorough bechmarking studies to catalog the accuracy and expense of molecular mechanical, semi-empirical and *ab initio* quantum mechanical methods.[44] Machine learning researchers customarily use datasets such as MNIST[45] and CIFAR-10[46] to contextualize the performance of novel algorithms. These benchmarks have provided a common ground to compare different approaches and quantitatively measure progress. The chemical machine learning community has thus built similar benchmark datasets, such as MoleculeNet,[47] the QMx series,[48–53] and the Therapeutics Data Commons,[54,55] among others.[56] Specialized benchmarks for generative models and inverse molecular design have also been developed, including GuacaMol,[57] Moses,[58] PMO,[59] and Tartarus[60].

Benchmarking of optimization algorithms has also received significant attention, with various frameworks having been established, like Coco[61] for continuous-parameter global optimization, OpenAI Gym[62] for reinforcement learning, Sherpa[63] and Optuna[64] for machine learning hyperparameter optimization, and Pygmo[65] for massively parallel optimization. Similar benchmarking frameworks, yet tailored to chemistry and materials science applications, have also been proposed. Summit[66] and EDBO+[14,15] are both concerned with the model-based optimization of chemical reactions. Olympus has attempted instead to cover a broader set of optimization tasks and algorithms. Both Olympus and

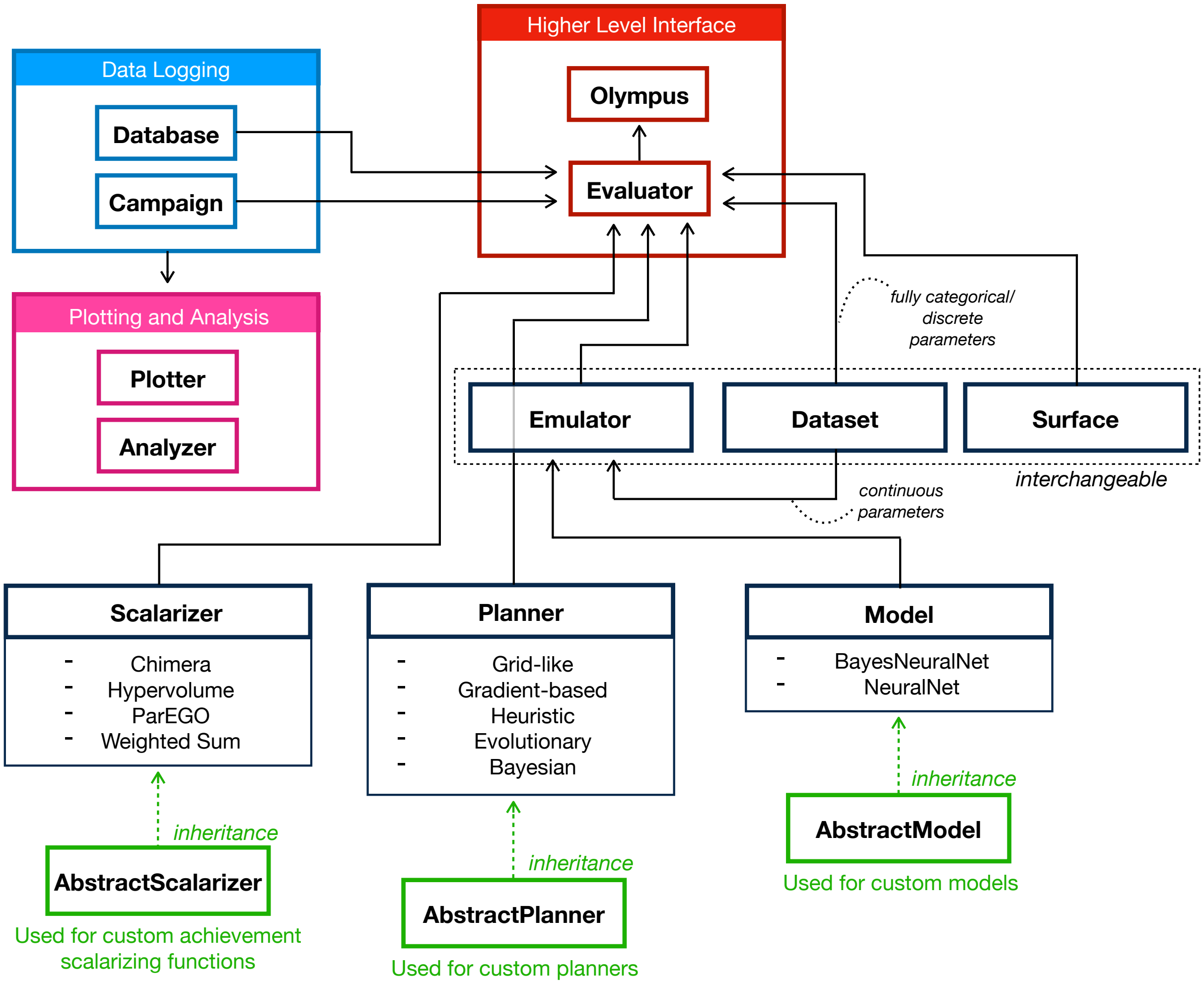


FIG. 1. Flow chart of dependence and inheritance for the core classes within the Olympus package.

Summit use probabilistic machine learning models to learn continuous parameter-measurement relationships from experimental datasets. This approach allows one to obtain simulated experimental measurements that can be used to benchmark optimization strategies. Without the burden of executing actual experiments, one can collect the necessary statistics to evaluate the expected performance of different algorithms.

III. PACKAGE UPDATES

A. Discrete, categorical and ordinal parameter types

The design of molecules, materials, and chemical processes necessitates the optimization of continuous, discrete, and categorical parameter types. For instance, when optimizing a catalyzed chemical reaction, one might want to select the catalyst ligand and solvent (categorical parameters) while also tuning the reaction time, temperature, and molarity of each component (continuous parameters).

Discrete parameters discretize a continuous interval into a finite number of options. They are characterized by an upper and lower bound, as well as a stride or step size which determines the resolution of the discretization. In Olympus, one instantiates a custom discrete parameter in one of two ways: (1) by defining an interval and a stride argument, resulting in evenly spaced options over that interval, or (2) by explicitly providing an ordered list of options.

```
from olympus.objects import ParameterDiscrete

# option #1: evenly spaced options
# over specified interval
param0 = ParameterDiscrete(
    name='param0',
    low=0.0,
    high=1.0,
    stride=0.1,
)
```

```
# option #2: list of options
param1 = ParameterDiscrete(
    name='param1',
    options=[0.1, 0.3, 0.7, 0.9],
)
```

Categorical parameters take one of a fixed number of possible options. The options by default have no inherent ordering. A commonly encountered categorical parameter in chemical reaction optimization is the molecular identity of the catalyst. Categorical parameters are defined according to a list of string options.

```
from olympus.objects import ParameterCategorical
param0 = ParameterCategorical(
    name='param0',
    options=['option0', 'option1', 'option2'],
)
```

In this case, categorical parameters are encoded as one-hot vectors. We also provide the opportunity to specify descriptors for each categorical option as lists of floats. Descriptors can be used to impose an ordering between the options of a categorical variable, and have been shown to accelerate the optimization rate of an experiment planning strategy if there exists sufficient correlation between descriptors and the property being optimized.[67]

```
from olympus.objects import ParameterCategorical
param0 = ParameterCategorical(
    name='param0',
    options=['option0', 'option1', 'option2'],
    descriptors=[[0.2, 0.8], [0.5, 0.5], [0.8, 0.3]],
)
```

As Olympus now provides continuous, discrete, and categorical parameter types, users can define arbitrarily complex parameter space objects using a combination of all three parameter types via a flexible interface.

```
from olympus.objects import (
    ParameterContinuous,
    ParameterDiscrete,
    ParameterCategorical
)
from olympus.campaigns import ParameterSpace

param0 = ParameterContinuous(
    name='param0',
    low=0.0,
    high=1.0,
)
param1 = ParameterDiscrete(
    name='param1',
    low=0.0,
    high=1.0,
    stride=0.1,
)
param2 = ParameterCategorical(
    name='param2',
    options=['option0', 'option1', 'option2'],
    descriptors=[[0.2, 0.8], [0.5, 0.5], [0.8, 0.3]],
)
param_space = ParameterSpace()
param_space.add(param0)
param_space.add(param1)
param_space.add(param2)
```

Finally, ordinal parameters are parameter sets with a natural ordering but the distances between options are unknown. Ordinal parameters are often useful to describe objective values (*i.e.*, experimental outcomes) that are qualitative in nature. For example, the result of a crystallization experiment may be qualitatively characterized (from least to most desirable) as providing a clear solution, a fine powder, or crystals of small/large size. Within Olympus, ordinal parameters are conventionally defined by ordering the options from least to most desirable. Additional information on the use of ordinal parameters within Olympus can be found in SI Sec. S.3.

```
from olympus.objects import ParameterOrdinal
param0 = ParameterOrdinal(
    name='param0',
    options=['worst', 'better', 'best'],
)
```

B. Multi-objective optimization with achievement scalarizing functions

The design of molecules, materials and chemical processes often requires the simultaneous optimization of multiple, competing objectives. Many solutions to the problem of concurrently optimizing multiple objective functions have been proposed.[68] A common approach is to construct a single objective function from multiple ones, by considering user-provided preferences about the optimization goal for each objective. Functions that map multiple separate objectives into a single, aggregate one are known as achievement scalarizing functions (ASFs). Ideally, the optimal solution of ASFs should correspond to a Pareto optimal solution of the original multi-objective problem.

We provide wrappers for four commonly used ASFs in the `Scalarizer` module of Olympus. The strategies are Weighted Sum,[69,70] Chebyshev,[71,72] Chimera,[73] and Hypervolume.[74–77] For brevity, we omit a detailed discussion of each ASF in the main text (details in SI Sec. S.5), and focus instead on their usage in Olympus. In particular, here we show an example using Chimera. One can instantiate the `Scalarizer` object using the following code.

```
from olympus.datasets import Dataset
from olympus.scalarizers import Scalarizer

dataset = Dataset(kind='suzuki_a')
scalarizer = Scalarizer(
    kind='Chimera',
    value_space=dataset.value_space,
    goals=['max', 'max'],
    tolerances=[0.8, 1.0],
    absolutes=[True, True],
)
```

Assume an optimization problem consists of parameter space $\mathcal{X} \in \mathbb{R}^d$ with d continuous parameters and an objective space $\mathcal{Y} \in \mathbb{R}^n$ with n objectives, corresponding to objective functions $\boldsymbol{f} = \{f_i\}_{i=1}^n : \mathcal{X} \mapsto \mathcal{Y}$. After the k^{th} optimization iteration, the `scalarize` method of the `Scalarizer` object takes as input a set of objective measurements $\{\boldsymbol{y}_i\}_{i=1}^k$ and applies the ASF, $s : \mathcal{Y} \mapsto [0, 1]$, to the current observations, returning a set of scalar-valued merits $\{\tilde{y}_i\}_{i=1}^k$. Conventionally, lower merit values correspond to a more optimal experiment. The scalarized dataset of observations $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^k$ is then passed to the experiment planning algorithm in order for the next parameter recommendations to be computed. Algorithm 1 shows pseudocode for multi-objective optimization with ASFs using the lower-level "ask-tell" interface of Olympus. Alternatively, the higher-level `Evaluator` interface for multi-objective ASF optimization is depicted in the code snippet below. After instantiating the `Scalarizer` object, it is passed to the `Evaluator`, along with the emulator, campaign and experiment planner instances. Effectively, the `Evaluator` abstracts away the "ask-tell" operations associated with an optimization experiment.

```
from olympus.emulators import Emulator
from olympus.scalarizers import Scalarizer
from olympus.planners import Planner
from olympus.evaluators import Evaluator
from olympus.campaigns import Campaign

emulator = Emulator(
    dataset='suzuki_a', model='BayesNeuralNet',
)
campaign = Campaign()
campaign.set_param_space(emulator.param_space)
campaign.set_value_space(emulator.value_space)

planner = Planner(kind='Gryffin')
planner.set_param_space(emulator.param_space)

scalarizer = Scalarizer(
    kind='Hypervolume',
    value_space=emulator.value_space,
    goals=['max', 'max'],
)

evaluator = Evaluator(
    dataset=emulator,
    campaign=campaign,
    planner=planner,
    scalarizer=scalarizer,
)
evaluator.optimize(num_iter=50)
```

Algorithm 1: Pseudocode for `ask-tell` interface multi-objective optimization using ASFs in Olympus.

```
Data: parameter space $\mathcal{X} \in \mathbb{R}^d$, objective space
      $\mathcal{Y} \in \mathbb{R}^n$, objective functions $\boldsymbol{f} : \mathcal{X} \mapsto \mathcal{Y}$, ASF
      $s : \mathcal{Y} \mapsto [0, 1]$, experiment planner, optimization
      budget, $b$
Result: dataset of observations $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^b$
        and scalarized observations $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^b$
$\mathcal{D} \leftarrow \emptyset$ ;
$\tilde{\mathcal{D}} \leftarrow \emptyset$ ;
$n_{eval} \leftarrow 0$ ;
while $n_{eval} < b$ do
    planner ← planner.tell($\tilde{\mathcal{D}}$) ;
    $\boldsymbol{x}_{next}$ ← planner.ask( ) ;
    $\boldsymbol{y}_{next} \leftarrow \boldsymbol{f}(\boldsymbol{x}_{next})$ ;
    $\mathcal{D} \leftarrow \mathcal{D} \cup (\boldsymbol{x}_{next}, \boldsymbol{y}_{next})$ ;
    $\tilde{\mathcal{D}}$ ← scalarizer.scalarize($\mathcal{D}$) ;
    $n_{eval} \leftarrow n_{eval} + 1$ ;
end
Function scalarize($\mathcal{D}$):
    $\{\tilde{y}_i\}_{i=1}^{|\mathcal{D}|} \leftarrow s\left(\{\boldsymbol{y}_i\}_{\boldsymbol{y}_i \in \mathcal{D}}\right)$ ;
    $\tilde{\mathcal{D}} \leftarrow \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{|\mathcal{D}|}$ ;
return $\tilde{\mathcal{D}}$
```

C. New analytical benchmark surfaces

The new version of Olympus ships with an updated suite of analytic benchmark surfaces to support the new discrete and categorical parameter types. Namely, we include 5 fully-categorical surfaces, `cat_camel`, `cat_michalewicz`, `cat_dejong`, `cat_slope` and `cat_ackley`. The parameter dimensionality and number of options per categorical variable can be user-specified. Fully-categorical surfaces are explained in greater detail and visualized in SI Sec. S.2 and SI Fig. S7, respectively. Also, given the software's extension to include multi-objective campaigns, we provide wrappers for 5 analytical benchmark surfaces with multiple objectives. Specifically, wrappers for the `mult_fonseca`, `mult_viennet`, `mult_zdt1`, `mult_zdt2`, `mult_zdt3` functions are provided, which comprise 2, 3, 2, 2, and 2 objectives, respectively. The multi-objective surfaces are described in further detail in SI Sec. S.2 and visualized in SI Fig. S8.

D. New datasets

Olympus provides 23 additional datasets adapted from the chemistry and material science literature. Datasets (referred to by the Olympus `kind` keyword) and specifications thereof are listed in Table I. We provide datasets from a diverse range of scientific disciplines, including chemical reaction optimization, materials manufacturing, organic photovoltaics, and energy storage. New datasets consist of between 3-10 parameters and 1-3 objectives. Several datasets feature exclusively categorical parameters (denoted by "cat" in the param types column). For these datasets, Olympus returns by default noisy measurements whose mean values are the discrete dataset values themselves. Individual measurements are sampled from Gaussian distributions around these values parameterized by standard deviations derived from Bayesian neural networks trained on these datasets. This approach constitutes a heteroscedastic noise model which intends to replicate the aleatoric uncertainty (inherent data noise) in the original measurements. More information on the noise model for fully-categorical datasets is given in SI Sec. S.4. In the fully-categorical cases, the `Dataset` and `Emualtor` objects are interchangeable, and can both be queried for measurements using their respective `run()` methods. For example, consider the fully-categorical dataset `perovskites`, which reports DFT computed bandgaps for hybrid organic-inorganic perovskite materials.[82]

```
from olympus.objects import ParameterVector
from olympus.datasets import Dataset

dataset = Dataset(kind='perovskites')
param = ParameterVector().from_dict(
    {
        'organic': 'ethylammonium',
        'cation': 'Ge',
        'anion': 'F',
    }
)
measurement = dataset.run(param)
print(measurement)
>>> 2.7138
```

Several datasets, including `perovskites`, ship with descriptors for categorical variables. Descriptors can be used to introduce ordering between categorical variable options. While descriptors have been shown to increase the optimization rate of experiment planners,[67] not all experiment planning algorithms in Olympus are compatible with descriptor-based representations. Table II lists the capabilities of experiment planners in Olympus. The constructor method of descriptor-capable planners has a `use_descriptors` argument which allows the user to toggle between a descriptor-based and a naïve one-hot-encoded representation of categorical variables. The code snippet below reuses the `perovskites` dataset instance.

```
from olympus.planners import Botorch

# print descriptors to console
descriptors = dataset.descriptors
print(descriptors.head())
>>>
     param         option         name     value
0  organic  ethylammonium         homo   -0.4601
1  organic  ethylammonium         lumo  -0.22398
2  organic  ethylammonium       dipole    1.3965
3  organic  ethylammonium  atomization  -1.84142
4  organic  ethylammonium        r_gyr  1.261565

# instantiate Botorch planner with descriptor-based
# representation
planner = Botorch(
    goal='minimize',
    use_descriptors=True,
)
```

Several of the new datasets contain continuous-valued parameters, and therefore require an emulator to return virtual measurements at all possible parameter settings. Among these datasets, `suzuki_a` through `suzuki_d` also comprise multiple objectives. We extended the `Emulator` module in Olympus to be compatible with multi-output regression such that one neural network model provides virtual measurements for all objectives. More information on the implementation and performance of new emulators is provided in SI Sec. S.3.

E. New and updated experiment planning algorithms

Olympus provides several updated and new experiment planning algorithms. We extend wrapper compatibility of the Random Search, Grid Search, Genetic,[105,120] HyperOpt,[112,121] and GPyOpt[111] planners to encompass discrete, categorical, and mixed-parameter types. We also added wrappers for 5 Bayesian optimization strategies, each compatible with all parameter types. The new algorithms are Gryffin,[67] SMAC,[116] Dragonfly,[117] HEBO,[119] and BoTorch.[118]

Gryffin[67] is an extension of the Phoenics[115] algorithm, which employs a kernel regression surrogate model, and can be applied to mixed continuous-categorical parameter spaces. SMAC[116] is a Bayesian optimization package that uses random forests as the surrogate model. Dragonfly[117] is an open source Python library for scalable and robust Bayesian optimization using Gaussian process (GP) surrogate models. Notably, Dragonfly allows for optimization over high-dimensional domains, optimization over structured combinatorial spaces and methods for handling parallel evaluations. The Dragonfly wrapper in Olympus does not currently support multi-fidelity optimization, parallel evaluations, or constrained optimization.

TABLE I. New datasets available in Olympus. Datasets which have a ✓in the emulator column contain continuous parameters, and are therefore emulated using Bayesian neural networks. Those which have an × are fully categorical datasets, and objective measurements consists of lookup in a table plus learned noise model. Datasets with (desc) in the param types column ship with descriptors for categorical variable options. More details on the datasets can be found in SI Sec. S.1.

Dataset	Research topic	# param	# obj	param types	# data points	Emulator
`diffvap_crystal`[19,78]	perovskite crystallization	10	1	cat (desc), disc, cont	918	✓
`dye_lasers`[79,80]	organic photophysics	3	3	cat	3468	×
`redoxmers`[81]	energy storage	4	3	cat (desc)	1408	×
`perovskites`[82]	HOIP solar cells	3	1	cat (desc)	192	×
`oer_plate_{a,b,c,d}`[83,84]	electrocatalysis	6	1	cont	2119-2121	✓
`p3ht`[85]	composite blends	5	1	cont	178	✓
`agnp`[86]	silver nanoparticles	5	1	cont	164	✓
`thin_films`[87]	thin film perovskites	3	1	cont	94	✓
`crossed_barrel`[27]	mechanical properties	4	1	cont, disc	600	✓
`autoam`[88]	3D printed structure	4	1	cont	100	✓
`suzuki_{i,ii,iii,iv}`[10]	organic chemistry	3	2	cont, cat	89-92	✓
`suzuki_edbo`[14,89]	organic chemistry	5	1	cat (desc)	3696	×
`buchwald_{a,b,c,d,e}`[14,90]	organic chemistry	4	1	cat (desc)	792	×

TABLE II. Capabilities of experiment planning strategies in Olympus. Planners are presented with their capabilities in terms of parameter types (all strategies support continuous parameters), multi-objective optimization compatibility via achievement scalarizing functions, support for descriptors of categorical variable options, and batched recommendations. For the Bayesian class of planners, the model type used for the surrogate is indicated, where TPE=Tree of parzen estimators, GP=Gaussian process, KDE=Kernel density estimation, and RF=Random forest. † We indicate strictly the compatibility of planners with multi-objective optimization *within the* Olympus *package*. Incompatibility therefore does not necessarily mean there does not exist an extension of the strategy to multi-objective problems elsewhere.

Planner	Class	Batched	Discrete	Categorical	Mixed	Multi-objective (*via* ASFs) †	Descriptors
Random Search	Grid-like	✓	✓	✓	✓	✓	×
Sobol Sequence[91]	Grid-like	✓	×	×	×	✓	N/A
Grid Search[92–94]	Grid-like	✓	✓	✓	✓	✓	×
Latin Hypercube[92–94]	Grid-like	✓	✓	×	×	✓	N/A
Steepest Descent[95,96]	Gradient-based	×	×	×	×	×	N/A
Conjugate Gradient[96,97]	Gradient-based	×	×	×	×	×	N/A
LBFGS[98–100]	Gradient-based	×	×	×	×	×	N/A
SLSQP[101]	Gradient-based	×	×	×	×	×	N/A
Snobfit[102]	Heuristic	×	×	×	×	×	N/A
Basin Hopping[103]	Heuristic	×	×	×	×	×	N/A
Simplex[104]	Heuristic	×	×	×	×	×	N/A
Genetic[105]	Evolutionary	✓	✓	✓	✓	✓	×
CMA-ES[106,107]	Evolutionary	✓	×	×	×	×	N/A
Particle Swarms[108,109]	Evolutionary	✓	×	×	×	×	N/A
Differential Evolution[110]	Evolutionary	✓	×	×	×	×	N/A
GPyOpt[111]	Bayesian (GP)	✓	✓	✓	✓	✓	×
HyperOpt[112–114]	Bayesian (TPE)	✓	✓	✓	✓	✓	×
Phoenics[115]	Bayesian (KDE)	✓	✓	×	×	✓	N/A
Gryffin[67]	Bayesian (KDE)	✓	✓	✓	✓	✓	✓
SMAC[116]	Bayesian (RF)	✓	✓	✓	✓	✓	×
Dragonfly[117]	Bayesian (GP)	✓	✓	✓	✓	×	×
BoTorch[118]	Bayesian (GP)	✓	✓	✓	✓	✓	✓
HEBO[119]	Bayesian (GP)	✓	✓	✓	✓	✓	✓

HEBO[119] (Heteroscedastic and Evolutionary Bayesian Optimization) is a GP-based Bayesian optimizer which performs non-linear input and output warping, admits exact marginal log-likelihood optimization and is robust to the values of learned parameters.

BoTorch is an in-house implementation of a GP-based Bayesian optimizer that relies on the BoTorch[118] and GPyTorch[122] libraries. For fully continuous parameter spaces and fully categorical parameter spaces *with* descriptors we use the Matern5/2 kernel. For fully categorical parameter spaces *without* descriptors we use the Hamming distance kernel, *i.e.*, $k\left(\boldsymbol{x},\boldsymbol{x}'\right) = \exp\left[-\mathrm{dist}\left(\boldsymbol{x},\boldsymbol{x}'\right)/\ell\right]$, where ℓ is a lengthscale parameter and $\mathrm{dist}\left(\boldsymbol{x},\boldsymbol{x}'\right) = 0$ if $\boldsymbol{x} = \boldsymbol{x}'$ and 1 if $\boldsymbol{x} \neq \boldsymbol{x}'$. For mixed continuous-categorical parameter spaces, we use a mixed kernel, which combines a categorical kernel based on Hamming distances and a Matern5/2 kernel, *i.e.*, $k(\boldsymbol{x}, y) = k_{\mathrm{cat}}(\boldsymbol{x}_{\mathrm{cat}}, y_{\mathrm{cat}}) \times k_{\mathrm{cont}}(\boldsymbol{x}_{\mathrm{cont}}, y_{\mathrm{cont}}) + k_{\mathrm{cat}}(\boldsymbol{x}_{\mathrm{cat}}, y_{\mathrm{cat}}) + k_{\mathrm{cont}}(\boldsymbol{x}_{\mathrm{cont}}, y_{\mathrm{cont}})$. Automatic relevance detection is used for the categorical kernel, *i.e.*, each categorical variable has an independent lengthscale parameter. By default, the expected improvement acquisition function is used.

The Olympus framework is also extended to support batched optimization, in which more than one parameter recommendation may be requested at once. However, this feature is limited to certain planners. Table II provides full details on the compatibility of Olympus planners with all new features.

F. New plotting and analysis

We made substantial updates to the analysis and plotting modules of our package. Table S29 lists and describes the supported plot types which can be accessed via the `Plotter` module of Olympus. In summary, we support three main plot types: traces, boxplots, and scatterplots. Trace plots compare the optimization performance of experiment planners by showing the cumulative best objective function value, regret value, or candidate rank achieved by an experiment planner as a function of the number of transpired objective function evaluations. We also provide trace plots which depict the cumulative fraction of top-k candidates evaluated by the experiment planner. As an example, consider an optimization using the `perovskites` dataset, where the goal is to minimize the bandgap across a dataset of hybrid organic-inorganic perovskite materials. Upon completion of the optimization benchmark (here comparing Random Search and Bayesian optimization via the BoTorch planner), we can ask Olympus to visualize the fraction of top-10 perovskite materials discovered by each strategy as a function of the number of bandgap evaluations. For brevity, we assume we already have an Olympus database instance called `database_perovskites` which contains the benchmark results. Figure 2 shows the plot generated by the following code snippet.

```python
from olympus import Plotter

plotter = Plotter()
plotter.plot_from_db(
    kind='traces_fraction_top_k',
    database=database_perovskites,
    threshold=10,
)
```

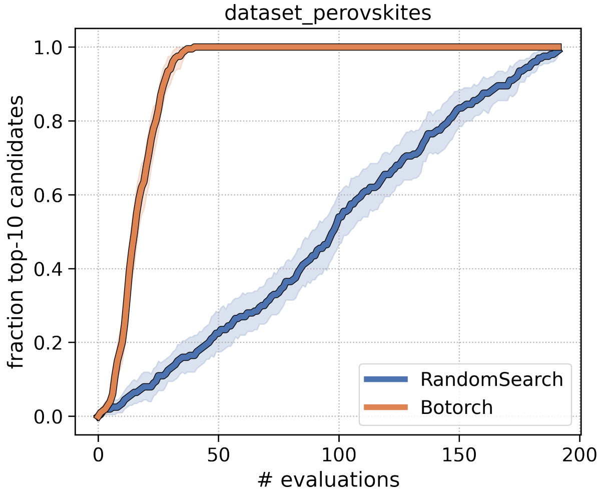


FIG. 2. Plot comparing the fraction of top-10 perovskite materials discovered with two experiment strategies as a function of the number of property evaluations. Solid traces show mean values over 40 independently seeded runs. Shaded region depict the 95% confidence interval.

Boxplots summarize the results of an optimization benchmark in a concise manner, showing the distribution of some performance metric for a series of experiment planning strategies over multiple executions. In particular, summarizing the relative performance of multiple experiment planning strategies on a multi-objective optimization problem (*i.e.*, the comparison of Pareto fronts) using a single scalar value is a valuable tool for a researcher to have available at their fingertips. One such metric is the hypervolume indicator, a set-quality indicator which measures the volume of the dominated portion of the objective space.[74–77] This metric is of great interest in multi-objective optimization research as it maintains the desirable feature of strict Pareto compliance. We omit a detailed introduction to the hypervolume indicator, and refer the interested reader to other works,[74–77] as well as SI Sec. S.5. In short, the hypervolume can be interpreted as the "size of the dominated space" and thus, experiment planning strategies which achieve the largest relative hypervolume can be considered the best performers. As an example, consider a multi-objective optimization experiment on the `mult_fonseca` analytic benchmark surface. Olympus produces boxplots of the normalized hypervolume over repeated executions of the experiment. Figure 3 shows the plot produced by the following code snippet.

```python
from olympus import Plotter

plotter = Plotter()
plotter.plot_from_db(
    kind='hypervolume',
    database=databse_multfonseca,
)
```

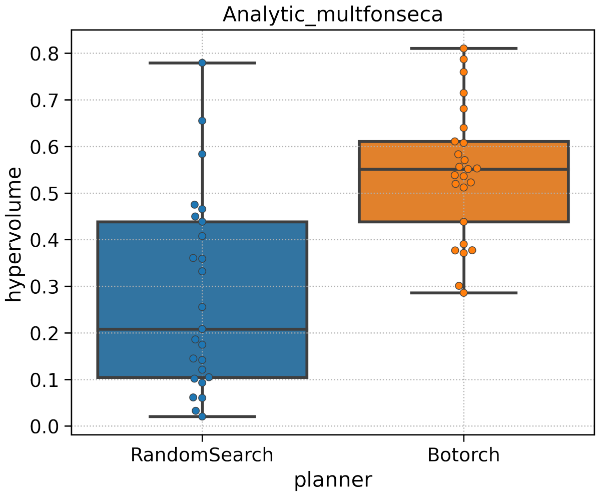


FIG. 3. Boxplot of hypervolume values for two experiment planners on the `mult_fonseca` surface. Boxplots show results over 25 independent runs of each experiment planner.

Olympus can also produce scatterplot visualizations of the Pareto front itself. This scatterplot can be accessed only for problems with 2 or 3 objectives. As an example, consider a comparison of Random Search and BoTorch on the `mult_fonseca` surface. Both objectives of this surface are to be minimized.

```python
from olympus import Plotter

plotter = Plotter()
plotter.plot_from_db(
    kind='pareto_front',
    database=databse_multfonseca,
)
```

G. Olympus web application

Web-based applications allow a piece of software to be accessed through a web-browser, removing the need for installation and the barrier to entry for users with limited programming experience. For example, Chemprop[123–125] is a Python package focusing on molecular property prediction that includes a web prediction interface allowing code-free access to trained models.[126] Torres *et al.* introduced EDBO+,[14,15] a package dedicated to multi-objective optimization of chemical reactions that features a web-based graphical user interface democratizing the software for non-experts.

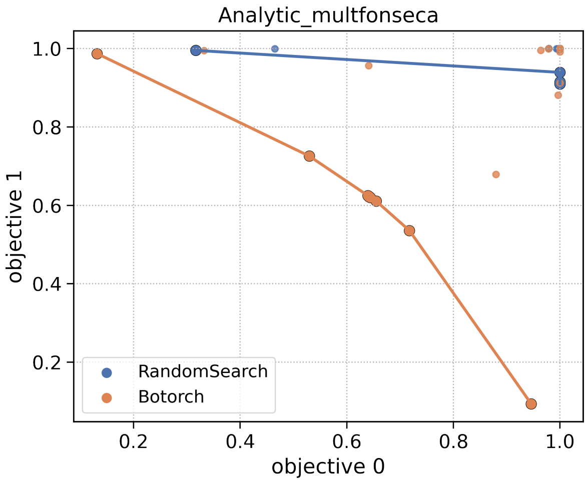


FIG. 4. Plot of the Pareto fronts identified by the Random Search and BoTorch experiment planners for the 2-dimensional Fonseca-Fleming surface (2 objectives, both with minimization goals). Larger points outlined in black indicate the Pareto front for each planner, connected by a line of the corresponding color.

To afford researchers with little programming experience the benefits of Olympus, we lay the foundation for code-free access to our package by constructing a web-based application using Streamlit. The Olympus web application will be made publicly accessible upon publication of this manuscript in a peer-reviewed journal.

Briefly, our webapp allows users to

- Conduct full benchmark experiments using any of the existing surfaces, datasets and planners in the package.
- Download datasets and descriptors in convenient csv format.
- Use the plotting and analysis modules to produce and download visualizations and summary statistics of their benchmark results.
- Upload their own datasets to be considered for addition to the Olympus package (submissions will first be reviewed by the developers).

IV. CASE STUDIES

To demonstrate the scope of research activities accessible with the updated version of Olympus, we perform two case studies that highlight (i) the performance of categorical Bayesian optimization, and (ii) the effect of ASF choice on optimization outcome.

A. Categorical optimization of chemical reaction conditions

Our first case demonstrates the ease with which one can conduct benchmark experiments for optimization of categorical parameter datasets using the high-level Olympus interface. This constitutes a minimal-code example where all operations of the benchmarking experiment are handled by the highest-level Olympus core class (Figure 1). As an expository dataset, we select the `suzuki_edbo` dataset, which reports yields for a Suzuki-Miyaura coupling reaction preformed on the nanomole scale using a automated flow-based synthesis platform.[14,89] The reaction scheme, along with the search space is presented in Figure 5a. This dataset consists of 5 categorical parmeters: the boronic acid derivative electrophile, aryl halide nucleophile, base used in the deprotonation step, Pd catalyst ligand, and solvent. Collectively, there are 3696 unique reactions.

With Olympus, one can rapidly construct a minimal source code optimization benchmark experiment consisting of experiment planning algorithms of type grid-like, evolutionary, and Bayesian. Namely, this demonstration uses Random Search, Genetic, HyperOpt, and BoTorch. Using the benchmark method of the main Olympus orchestrator, one simply needs to pass the name of the dataset, a list of experiment planner names, an Olympus database instance to store the results, and some specifications pertaining to the length of the benchmark experiment. The `num_ind_runs` argument specifies the number of repeated executions of each planner, while the `num_iter` argument specifies the number of objective function evaluations to perform in each run. Source code for this example appears in Figure 5b.

Olympus' plotting module then provides users with access to a series of visualizations of benchmark results. Here, we choose the `num_evals_top_k` type boxplot, which constructs distributions of the number of yield evaluations needed for each experiment planner to identify a top-20 yield with respect to all 3696 measurements. For this dataset, a top-20 value roughly corresponds to a yield of $\geq 96\%$. The resulting boxplot is shown in Figure 5c. It is immediately clear that the Bayesian optimizers (HyperOpt and BoTorch) perform significantly better than Random Search and Genetic on this task. HyperOpt is able to identify a top-20 yield on average within 79 ± 10 evaluations, while BoTorch is able to do the same within only 27 ± 2 evaluations. Ultimately, the insight gleaned from this case study should inform choices of experiment planning algorithm in future experiments.

B. Multi-objective design of redox-active materials

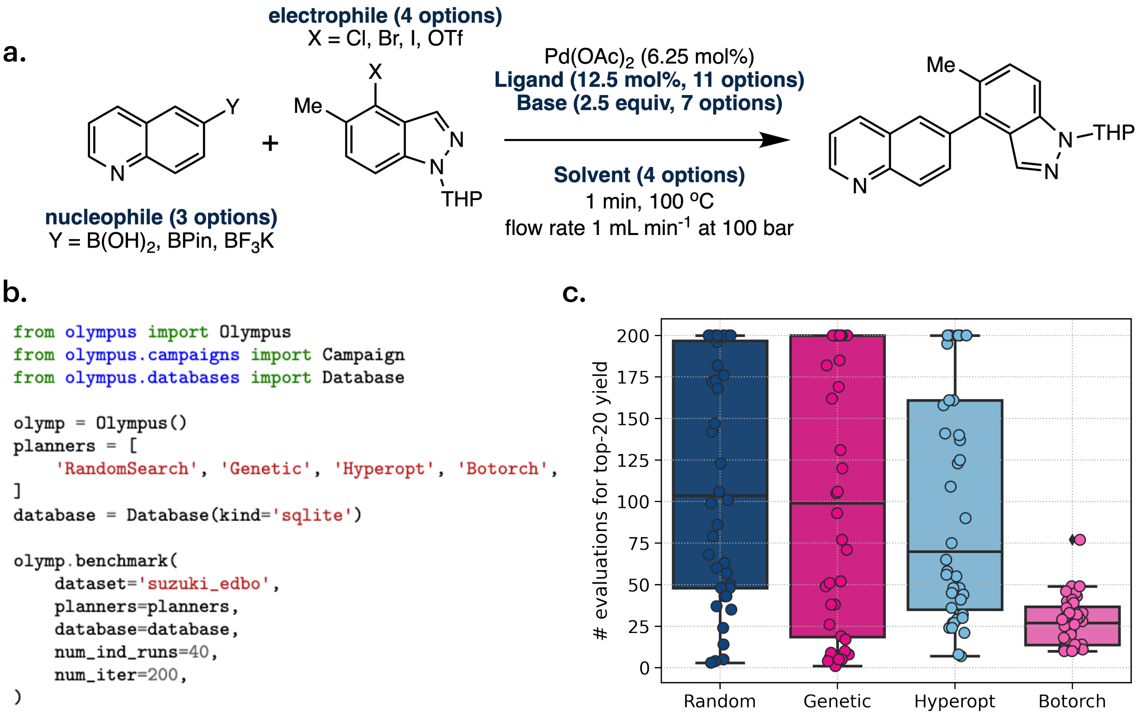
The choice of ASF can have profound effects on the result of an optimization campaign. In fact, the Pareto optimal solution that is recommended by an experiment planner depends on the choice of ASF and its parameterization.

This case study demonstrates how one can evaluate the behavior of different ASFs in Olympus. We consider the `redoxmers` dataset, in which Agarwal *et al.*[81] reported a computational screen of 1408 benzothiadiazole derivatives for application as redox-active materials in non-aqueous flow batteries (Fig. 6a). The three objectives are the DFT-computed reduction potential E^{red}, solvation free energy G^{solv} and maximum absorption wavelength λ^{abs}. We optimize the three objectives simultaneously using the BoTorch planner and each ASF supported in Olympus. The aim is to identify candidates that (i) have an maximum absorption wavelength as close as possible to 375 nm, (ii) have a minimal reduction potential against a Li/Li^{+} reference electrode, and (iii) provide the lowest possible solvation free energy, understood as a proxy for solubility.

The Weighted Sum and Chimera ASF strategies require parameterization. Weighted Sum received weights of 3, 2, and 1 for the λ^{abs}, E^{red} and G^{solv} objectives, respectively, indicating the preferential order of objective importance. The objective hierarchy of Chimera was organized in the same order. Chimera also received absolute tolerances of 25 nm for the λ^{abs} objective, and 2.04 V for the E^{red} objective.

Results of this experiment are shown in Fig. 6c. Subplots show optimization traces over 200 evaluations corresponding to the redoxmer candidate objective values with the best scalarized merit. Each ASF campaign was repeated 40 times. Most notably, the hierarchy and absolute tolerances provided to Chimera modulate which property values are considered advantageous. Chimera sacrifices slightly the achieved value of the first objective compared to all other ASFs, and the value of the second objective compared to Hypervolume. As a consequence, Chimera is able to achieve a significantly lower G^{solv} than all other ASFs. Importantly, the Chimera solutions still satisfy tolerances on the first two objectives. The Weighted Sum strategy appears to suffer from the well-known difficulties associated with choosing weighting schemes that result in desired solutions[69,70] and over-prioritizes minimization of λ^{abs}, in turn restricting values of the latter two objectives to be sub-optimal. Hypervolume provides a well-balanced multi-objective optimization of the redoxmer candidates, and, unsurprisingly, generates Pareto fronts which have on average the largest dominated hypervolume (Fig. 7). The dominated hypervolume facilitates assessment of Pareto fronts by summarizing their characteristics (such as proximity to the Pareto front, diversity and spread) with a single scalar value.[74–77]

To reinforce that the choice of ASF modulates which



```
from olympus import Olympus
from olympus.campaigns import Campaign
from olympus.databases import Database

olymp = Olympus()
planners = [
    'RandomSearch', 'Genetic', 'Hyperopt', 'Botorch',
]
database = Database(kind='sqlite')

olymp.benchmark(
    dataset='suzuki_edbo',
    planners=planners,
    database=database,
    num_ind_runs=40,
    num_iter=200,
)
```

FIG. 5. a) Reaction scheme of the Pd-catalyzed Suzuki-Miyaura cross-coupling reaction reported by Perera *et al.*[127] The fully-categorical parameter space consists of varying nucleophile and electrophile substrates, catalyst ligand, base, and solvent. The full Cartesian product space consists of 3696 unique reactions. b) Minimal code required for running a benchmark optimization experiment using several planners for the reaction shown in a). c) Visualization of benchmark results using the plotting module of Olympus. Box-and-whisker plots show the number of evaluations needed for each strategy to achieve a top-20 yield value. Each of the 40 replicate optimization experiments are conducted for a maximum of 200 evaluations. For this dataset, a top-20 yield evaluation corresponds to a yield of $\gtrsim$ 96%.

redoxmer candidates are deemed desirable, we discuss the frequency at which each R-group option is featured in high performing candidates. Fig. 6b shows the relative frequency with which each R-group option is included in top-20 redoxmer candidates, as ranked by each ASF. The underpinnings of the ASFs, as well as the preference information encoded by the user often results in stark differences in the rate of occurrence of specific chemical substituents in candidates that are deemed desirable. For instance, top candidates according to Chimera and Hypervolume almost exclusively feature the ether substituent at the R_1 position (R_1^1), where as Weighted Sum and Chebyshev top candidates feature both the ether and perfluorinated substituents at roughly the same rate. Also noteworthy is the relatively strong preference of Chimera and Hypervolume for the dimethylamine group at the R_2 and R_3 positions ($R_{2,3}^8$), the preference of Weighted Sum and Chebyshev for R_4^8, and Hypervolume for R_4^{11}.

Ultimately, this case study demonstrates how the choice and parameterization of ASF for a multi-objective materials discovery campaign affects the properties (and therefore the structure) of the resulting candidates. Olympus provides users with the means to quickly develop intuition on how certain ASF choices might influence the outcome of data-driven experimental campaigns.

V. CONCLUSION

This work describes an extension of the Olympus Python package to include any combination of discrete, categorical, or ordinal parameters. We have also allowed for the definition and execution of multi-objective optimization campaigns by providing wrappers for several achievement scalarizing functions. To promote usage of these new features, we have provided access to 23 new datasets derived from chemistry and materials science literature, and 10 new analytical optimization benchmark surfaces, all of which feature either categorical parameters or multiple objectives. Several experiment planning algorithms have been updated to be compatible with the new parameter types, and 5 additional Bayesian optimization algorithms have been added. We improved the plotting and analysis capabilities of Olympus, enabling users to straightforwardly compare optimization campaigns using various metrics. Lastly, we have laid the foundations for programming-free access to the core functionality of Olympus by designing a web-based applica-

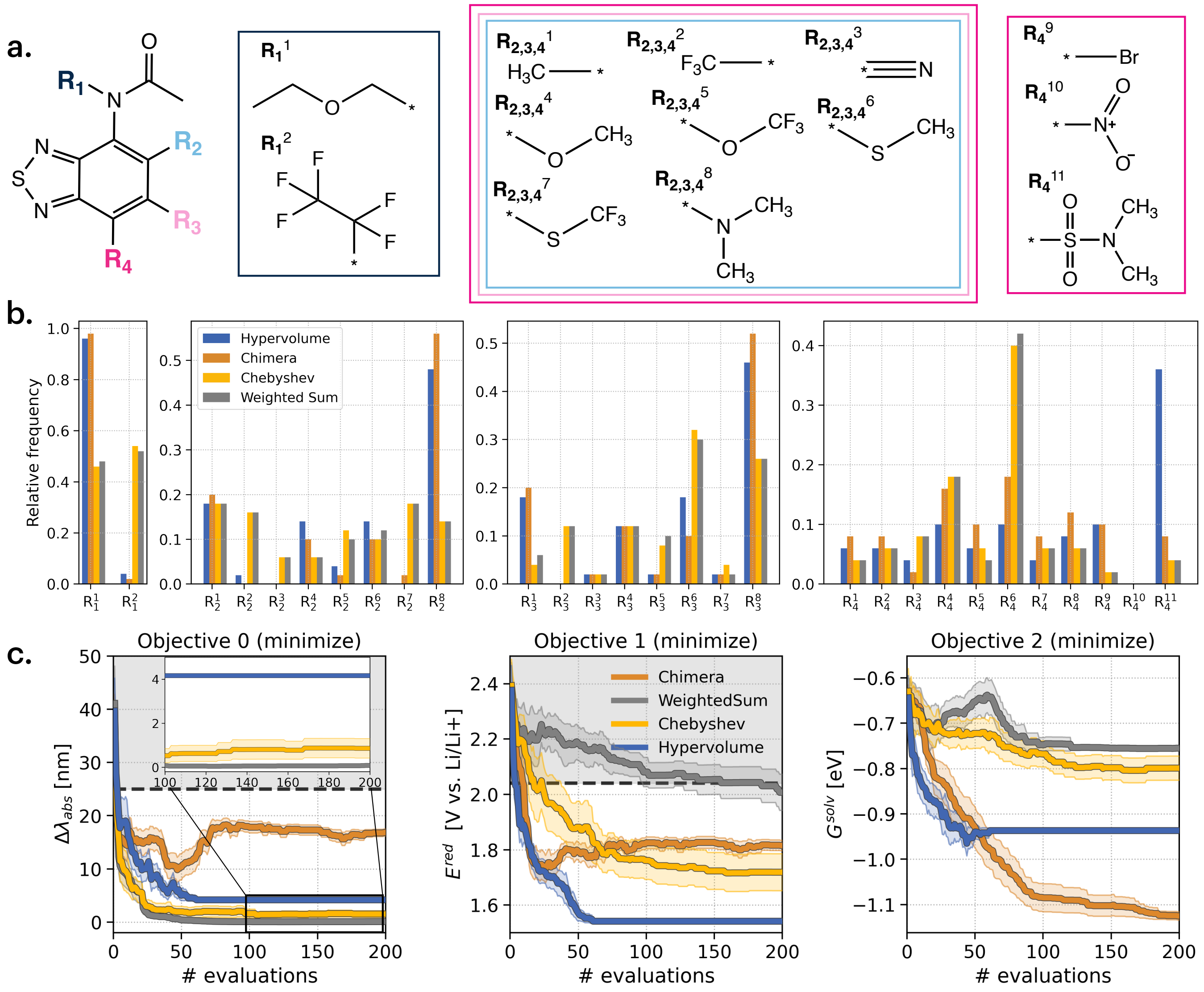


FIG. 6. Optimization setup and results for case study 2: multi-objective design of redox-active materials for non-aqueous redox flow batteries. (a) Markush structure of the benzothiadiazole scaffold and all substituents considered. The entire design space consists of 1408 candidates (2 R_1 × 8 R_2 × 8 R_3 × 11 R_4 options). (b) Relative frequency of substituent occurrence in the top 20 redoxmer candidates ranked using each ASF. (c) Optimization traces using the BoTorch planner with different ASFs. Traces depict the objective function values corresponding to the candidate with the best scalarized merit value at each iteration. Shaded bands indicate the 95% interval over 40 independent repeats. The gray shaded areas indicate regions in which the Chimera tolerance for that objective is not satisfied.

tion using Streamlit. The two case studies presented, on the categorical optimization of chemical reaction conditions and the multi-objective design of redox active materials, show how Olympus allows researchers to quickly benchmark numerous optimization strategies and gain insight into their expected behaviour on realistic tasks.

CODE AND DATA AVAILABILITY

Olympus is available in its entirety on GitHub (https://github.com/aspuru-guzik-group/olympus/tree/main) under an MIT license. All the code and data needed to reproduce the computational experiments in this manuscript are also included in this repository. Olympus is available for installation through PyPI. The Olympus web application will be made publicly accessible upon publication of this article in a peer-reviewed journal.

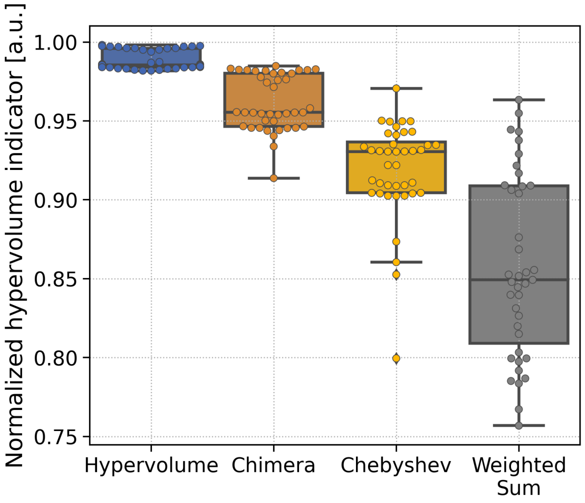


FIG. 7. Hypervolume indicator metric used to asses the quality and diversity of Pareto sets from the completed redoxmer design campaigns (*i.e.*, after 200 optimization evaluations). The common reference point is set as the worst (maximum) values for each objective. Values are normalized with respect to the largest dominated hypervolume value achieved across all campaigns.

ACKNOWLEDGMENTS

The authors thank Dr. Martin Seifrid, Dr. Felix Strieth-Kalthoff, and Sterling Baird for contribution to valuable discussion. R.J.H. gratefully acknowledges the Natural Sciences and Engineering Research Council of Canada (NSERC) for provision of the Postgraduate Scholarships-Doctoral Program (PGSD3-534584-2019), as well as support from the Vector Institute. A.A.-G. acknowledges support from the Canada 150 Research Chairs program and CIFAR, as well as the generous support of Anders G. Fröseth. This work relates to the Department of Navy award (N00014-19-1-2134) issued by the office of Naval Research. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research. Work by Q.A. and J.S. is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001118C0036. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA. J.S. also acknowledges resources of the MERCURY consortium (http://mercuryconsortium.org/) under NSF Grant No. CNS-2018427. Computations reported in this work were performed on the computing clusters of the Vector Institute and on the Niagara supercomputer at the SciNet HPC Consortium.[128,129] Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. SciNet is funded by the Canada Foundation for Innovation, the Government of Ontario, Ontario Research Fund - Research Excellence, and by the University of Toronto.

CONFLICTS OF INTEREST

A.A.-G is the Chief Visionary Officer and founding member of Kebotix, Inc. J.S. is a scientific advisor for Atinary Technologies, Inc.

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[2] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[4] A. Leach, *Molecular Modelling: Principles and Applications*. Prentice Hall, 2001.

[5] J. P. McMullen and K. F. Jensen, "An automated microfluidic system for online optimization in chemical synthesis," *Organic process research & development*, vol. 14, no. 5, pp. 1169–1176, 2010.

[6] D. E. Fitzpatrick, C. Battilocchio, and S. V. Ley, "A novel internet-based reaction monitoring, control and autonomous self-optimization platform for chemical synthesis," *Organic Process Research & Development*, vol. 20, no. 2, pp. 386–394, 2016.

[7] D. Cortés-Borda, E. Wimmer, B. Gouilleux, E. Barré, N. Oger, L. Goulamaly, L. Peault, B. Charrier, C. Truchet, P. Giraudeau, *et al.*, "An autonomous self-optimizing flow reactor for the synthesis of natural product carpanone," *The Journal of organic chemistry*, vol. 83, no. 23, pp. 14286–14299, 2018.

[8] B. E. Walker, J. H. Bannock, A. M. Nightingale, and J. C. deMello, "Tuning reaction products by constrained optimisation," *Reaction Chemistry & Engineering*, vol. 2, no. 5, pp. 785–798, 2017.

[9] S. Krishnadasan, R. Brown, A. Demello, and J. Demello, "Intelligent routes to the controlled synthesis of nanoparticles," *Lab on a Chip*, vol. 7, no. 11, pp. 1434–1441, 2007.

[10] L. M. Baumgartner, C. W. Coley, B. J. Reizman, K. W. Gao, and K. F. Jensen, "Optimum catalyst selection over continuous and discrete process variables with a single droplet microfluidic reaction platform," *Reaction Chemistry & Engineering*, vol. 3, no. 3, pp. 301–311, 2018.

[11] A. Schweidtmann, A. Clayton, N. Holmes, E. Bradford, R. Bourne, and A. Lapkin, "Machine learning meets continuous flow chemistry: Automated optimization towards the pareto front of multiple objectives," *Chem. Eng. J*, pp. 177–282, 2018.

[12] A.-C. Bédard, A. Adamo, K. C. Aroh, M. G. Russell, A. A. Bedermann, J. Torosian, B. Yue, K. F. Jensen, and T. F. Jamison, "Reconfigurable system for automated optimization of diverse chemical reactions," *Science*, vol. 361, no. 6408, pp. 1220–1225, 2018.

[13] M. Christensen, L. P. E. Yunker, F. Adedeji, F. Häse, L. M. Roch, T. Gensch, G. dos Passos Gomes, T. Zepel, M. S. Sigman, A. Aspuru-Guzik, and J. E. Hein, "Data-science driven autonomous process optimization," *Communications Chemistry*, vol. 4, pp. 1–12, Aug. 2021.

[14] B. J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J. I. M. Alvarado, J. M. Janey, R. P. Adams, and A. G. Doyle, "Bayesian reaction optimization as a tool for chemical synthesis," *Nature*, vol. 590, pp. 89–96, Feb. 2021.

[15] J. G. Torres, S. H. Lau, P. Anchuri, J. Stevens, J. Tabora, J. Li, A. Borovika, R. Adams, and A. Doyle, "A Multi-Objective Active Learning Platform and Web App for Reaction Optimization," Aug. 2022.

[16] A. M. K. Nambiar, C. P. Breen, T. Hart, T. Kulesza, T. F. Jamison, and K. F. Jensen, "Bayesian Optimization of Computer-Proposed Multistep Synthetic Routes on an Automated Robotic Flow Platform," *ACS Central Science*, vol. 8, pp. 825–836, June 2022. Publisher: American Chemical Society.

[17] 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*, vol. 6, no. 20, 2020.

[18] S. Langner, F. Häse, J. Perea, T. Stubhan, J. Hauch, L. Roch, T. Heumueller, A. Aspuru-Guzik, and C. Brabec, "Beyond ternary opv: High-throughput experimentation and self-driving laboratories optimize multicomponent systems," *Advanced Materials*, p. 1907801, 2020.

[19] N. T. P. Hartono, M. Ani Najeeb, Z. Li, P. W. Nega, C. A. Fleming, X. Sun, E. M. Chan, A. Abate, A. J. Norquist, J. Schrier, and T. Buonassisi, "Principled Exploration of Bipyridine and Terpyridine Additives to Promote Methylammonium Lead Iodide Perovskite Crystallization," *Crystal Growth & Design*, vol. 22, pp. 5424–5431, Sept. 2022. Publisher: American Chemical Society.

[20] P. Nikolaev, D. Hooper, F. Webber, R. Rao, K. Decker, M. Krein, J. Poleski, R. Barto, and B. Maruyama, "Autonomy in materials research: a case study in carbon nanotube growth," *npj Computational Materials*, vol. 2, no. 1, pp. 1–6, 2016.

[21] H. Tao, T. Wu, S. Kheiri, M. Aldeghi, A. Aspuru-Guzik, and E. Kumacheva, "Self-Driving Platform for Metal Nanoparticle Synthesis: Combining Microfluidics and Machine Learning," *Advanced Functional Materials*, vol. 31, no. 51, 2021.

[22] K. Vaddi, H. T. Chiang, and L. D. Pozzo, "Autonomous retrosynthesis of gold nanoparticles via spectral shape matching," *Digital Discovery*, vol. 1, pp. 502–510, Aug. 2022. Publisher: RSC.

[23] A. Deshwal, C. M. Simon, and J. Rao Doppa, "Bayesian optimization of nanoporous materials," *Molecular Systems Design & Engineering*, vol. 6, no. 12, pp. 1066–1086, 2021.

[24] R. J. Hickman, P. Bannigan, Z. Bao, A. Aspuru-Guzik, and C. Allen, "Self-driving laboratories: A paradigm shift in nanomedicine development," *Matter*, Mar. 2023.

[25] B. Rohr, H. S. Stein, D. Guevarra, Y. Wang, J. A. Haber, M. Aykol, S. K. Suram, and J. M. Gregoire, "Benchmarking the acceleration of materials discovery by sequential learning," *Chemical Science*, vol. 11, no. 10, pp. 2696–2706, 2020.

[26] B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes, *et al.*, "A mobile robotic chemist," *Nature*,

vol. 583, no. 7815, pp. 237–241, 2020.

[27] A. E. Gongora, B. Xu, W. Perry, C. Okoye, P. Riley, K. G. Reyes, E. F. Morgan, and K. A. Brown, "A bayesian experimental autonomous researcher for mechanical design," *Science Advances*, vol. 6, no. 15, p. eaaz1708, 2020.

[28] M. J. Tamasi, R. A. Patel, C. H. Borca, S. Kosuri, H. Mugnier, R. Upadhya, N. S. Murthy, M. A. Webb, and A. J. Gormley, "Machine Learning on a Robotic Platform for the Design of Polymer–Protein Hybrids," *Advanced Materials*, vol. 34, no. 30, p. 2201809, 2022.

[29] M. M. Noack, K. G. Yager, M. Fukuto, G. S. Doerk, R. Li, and J. A. Sethian, "A kriging-based approach to autonomous experimentation with applications to X-ray scattering," *Scientific reports*, vol. 9, no. 1, pp. 1–19, 2019.

[30] P. B. Wigley, P. J. Everitt, A. van den Hengel, J. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. Quinlivan, P. Manju, C. C. Kuhn, *et al.*, "Fast machine-learning online optimization of ultra-cold-atom experiments," *Scientific reports*, vol. 6, p. 25890, 2016.

[31] D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue, and T. Lookman, "Accelerated search for materials with targeted properties by adaptive design," *Nature communications*, vol. 7, no. 1, pp. 1–9, 2016.

[32] P. Bannigan, Z. Bao, R. J. Hickman, M. Aldeghi, F. Häse, A. Aspuru-Guzik, and C. Allen, "Machine learning models to accelerate the design of polymeric long-acting injectables," *Nature Communications*, vol. 14, p. 35, Jan. 2023.

[33] F. Häse, L. M. Roch, and A. Aspuru-Guzik, "Next-generation experimentation with self-driving laboratories," *Trends in Chemistry*, 2019.

[34] E. Stach, B. DeCost, A. G. Kusne, J. Hattrick-Simpers, K. A. Brown, K. G. Reyes, J. Schrier, S. Billinge, T. Buonassisi, I. Foster, C. P. Gomes, J. M. Gregoire, A. Mehta, J. Montoya, E. Olivetti, C. Park, E. Rotenberg, S. K. Saikin, S. Smullin, V. Stanev, and B. Maruyama, "Autonomous experimentation systems for materials development: A community perspective," *Matter*, vol. 4, pp. 2702–2726, Sept. 2021.

[35] K. F. Jensen, C. W. Coley, and N. S. Eyke, "Autonomous discovery in the chemical sciences part i: Progress," *Angewandte Chemie International Edition*, 2019.

[36] C. W. Coley, N. S. Eyke, and K. F. Jensen, "Autonomous discovery in the chemical sciences part ii: Outlook," *Angewandte Chemie International Edition*, 2019.

[37] M. M. Flores-Leonar, L. M. Mejía-Mendoza, A. Aguilar-Granda, B. Sanchez-Lengeling, H. Tribukait, C. Amador-Bedolla, and A. Aspuru-Guzik, "Materials Acceleration Platforms: On the way to autonomous experimentation," *Current Opinion in Green and Sustainable Chemistry*, vol. 25, p. 100370, Oct. 2020.

[38] H. S. Stein and J. M. Gregoire, "Progress and prospects for accelerating materials science with automated and autonomous workflows," *Chemical Science*, vol. 10, no. 42, pp. 9640–9649, 2019.

[39] J. Yano, K. J. Gaffney, J. Gregoire, L. Hung, A. Ourmazd, J. Schrier, J. A. Sethian, and F. M. Toma, "The case for data science in experimental chemistry: examples and recommendations," *Nat Rev Chem*, vol. 6, pp. 357–370, Apr. 2022.

[40] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[41] S. Droste, T. Jansen, and I. Wegener, "Optimization with randomized search heuristics—the (a)nfl theorem, realistic scenarios, and difficult functions," *Theoretical Computer Science*, vol. 287, no. 1, pp. 131 – 144, 2002. Natural Computing.

[42] F. Häse, M. Aldeghi, R. J. Hickman, L. M. Roch, M. Christensen, E. Liles, J. E. Hein, and A. Aspuru-Guzik, "Olympus: a benchmarking framework for noisy optimization and experiment planning," *Machine Learning: Science and Technology*, vol. 2, p. 035021, July 2021. Publisher: IOP Publishing.

[43] The Olympus authors, "Olympus: a benchmarking framework for noisy optimization and experiment planning." `https://github.com/aspuru-guzik-group/olympus`, 2020.

[44] R. A. Mata and M. A. Suhm, "Benchmarking Quantum Chemical Methods: Are We Heading in the Right Direction?," *Angewandte Chemie International Edition*, vol. 56, no. 37, pp. 11011–11018, 2017.

[45] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[46] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., 2009.

[47] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande, "Moleculenet: a benchmark for molecular machine learning," *Chemical science*, vol. 9, no. 2, pp. 513–530, 2018.

[48] L. C. Blum and J.-L. Reymond, "970 million druglike small molecules for virtual screening in the chemical universe database gdb-13," *Journal of the American Chemical Society*, vol. 131, no. 25, pp. 8732–8733, 2009.

[49] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17," *Journal of Chemical Information and Modeling*, vol. 52, no. 11, pp. 2864–2875, 2012.

[50] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Phys. Rev. Lett.*, vol. 108, p. 058301, Jan 2012.

[51] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific Data*, vol. 1, no. 1, p. 140022, 2014.

[52] R. Ramakrishnan, M. Hartmann, E. Tapavicza, and O. A. von Lilienfeld, "Electronic spectra from tddft and machine learning in chemical space," *The Journal of Chemical Physics*, vol. 143, no. 8, p. 084111, 2015.

[53] M. Glavatskikh, J. Leguy, G. Hunault, T. Cauchy, and B. Da Mota, "Dataset's chemical diversity limits the generalizability of machine learning predictions," *Journal of Cheminformatics*, vol. 11, no. 1, p. 69, 2019.

[54] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. Coley, C. Xiao, J. Sun, and M. Zitnik, "Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*

(J. Vanschoren and S. Yeung, eds.), vol. 1, 2021.

[55] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. W. Coley, C. Xiao, J. Sun, and M. Zitnik, "Artificial intelligence foundation for therapeutic science," *Nature Chemical Biology*, vol. 18, pp. 1033–1036, Oct. 2022. Number: 10 Publisher: Nature Publishing Group.

[56] L. Chanussot*, A. Das*, S. Goyal*, T. Lavril*, M. Shuaibi*, M. Riviere, K. Tran, J. Heras-Domingo, C. Ho, W. Hu, A. Palizhati, A. Sriram, B. Wood, J. Yoon, D. Parikh, C. L. Zitnick, and Z. Ulissi, "Open catalyst 2020 (oc20) dataset and community challenges," *ACS Catalysis*, 2021.

[57] N. Brown, M. Fiscato, M. H. Segler, and A. C. Vaucher, "Guacamol: Benchmarking models for de novo molecular design," *Journal of Chemical Information and Modeling*, vol. 59, no. 3, pp. 1096–1108, 2019.

[58] D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, A. Kadurin, S. Nikolenko, A. Aspuru-Guzik, and A. Zhavoronkov, "Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models," *arXiv preprint arXiv:1811.12823*, 2018.

[59] W. Gao, T. Fu, J. Sun, and C. W. Coley, "Sample efficiency matters: a benchmark for practical molecular optimization," *arXiv preprint arXiv:2206.12411*, 2022.

[60] A. Nigam, R. Pollice, G. Tom, K. Jorner, L. A. Thiede, A. Kundaje, and A. Aspuru-Guzik, "Tartarus: A benchmarking platform for realistic and practical inverse molecular design," 2022.

[61] O. Elhara, K. Varelas, D. Nguyen, T. Tusar, D. Brockhoff, N. Hansen, and A. Auger, "Coco: The large scale black-box optimization benchmarking (bbob-largescale) test suite," *arXiv preprint arXiv:1903.06396*, 2019.

[62] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[63] L. Hertel, J. Collado, P. Sadowski, J. Ott, and P. Baldi, "Sherpa: Robust hyperparameter optimization for machine learning," *arXiv preprint arXiv:2005.04048*, 2020.

[64] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

[65] F. Biscani and D. Izzo, "esa/pagmo2: pagmo 2.15.0," Apr. 2020.

[66] K. C. Felton, J. G. Rittig, and A. A. Lapkin, "Summit: Benchmarking Machine Learning Methods for Reaction Optimisation," *Chemistry–Methods*, vol. 1, no. 2, pp. 116–122, 2021.

[67] F. Häse, M. Aldeghi, R. J. Hickman, L. M. Roch, and A. Aspuru-Guzik, "Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge," *Applied Physics Reviews*, no. 8, p. 031406, 2021.

[68] K. Deb and K. Deb, "Multi-objective Optimization," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (E. K. Burke and G. Kendall, eds.), pp. 403–449, Boston, MA: Springer US, 2014.

[69] I. Y. Kim and O. L. de Weck, "Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation," *Structural and Multidisciplinary Optimization*, vol. 31, pp. 105–116, Feb. 2006.

[70] C. A. C. Coello, S. Gonz&aacute, L. Brambila, Josu&eacute, F. Gamboa, M. G. C. Tapia, R. Hern&aacute, N. G&oacute, and mez, "Evolutionary multiobjective optimization: open research areas and some challenges lying ahead.," *Complex &amp; Intelligent Systems*, vol. 6, pp. 221–237, July 2020. Publisher: Springer.

[71] J. Knowles and E. J. Hughes, "Multiobjective optimization on a budget of 250 evaluations," in *Evolutionary Multi-Criterion Optimization (EMO-2005)* (C. Coello et al, ed.), vol. 3410 of *LNCS*, Springer-Verlag, 2005.

[72] J. Knowles, "Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[73] F. Häse, L. M. Roch, and A. Aspuru-Guzik, "Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories," *Chemical Science*, vol. 9, no. 39, pp. 7642–7655, 2018.

[74] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms — A comparative case study," in *Parallel Problem Solving from Nature — PPSN V* (A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, eds.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 292–301, Springer, 1998.

[75] J. Knowles, D. Corne, and M. Fleischer, "Bounded archiving using the lebesgue measure," in *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, vol. 4, pp. 2490–2497 Vol.4, 2003.

[76] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *ACM Comput. Surv.*, vol. 52, mar 2019.

[77] A. P. Guerreiro, C. M. Fonseca, and L. Paquete, "The Hypervolume Indicator: Problems and Algorithms," *ACM Computing Surveys*, vol. 54, pp. 1–42, July 2021. arXiv:2005.00515 [cs].

[78] Z. Li, P. W. Nega, M. A. N. Nellikkal, C. Dun, M. Zeller, J. J. Urban, W. A. Saidi, J. Schrier, A. J. Norquist, and E. M. Chan, "Dimensional Control over Metal Halide Perovskite Crystallization Guided by Active Learning," *Chemistry of Materials*, vol. 34, pp. 756–767, Jan. 2022. Publisher: American Chemical Society.

[79] M. Seifrid, R. J. Hickman, A. Aguilar-Granda, C. Lavigne, J. Vestfrid, T. C. Wu, T. Gaudin, E. J. Hopkins, and A. Aspuru-Guzik, "Routescore: Punching the ticket to more efficient materials development," *ACS Central Science*, vol. 8, no. 1, pp. 122–131, 2022.

[80] M. Seifrid, R. J. Hickman, A. Aguilar-Granda, C. Lavigne, J. Vestfrid, T. C. Wu, T. Gaudin, E. J. Hopkins, and A. Aspuru-Guzik, "Code and Data for "Routescore: Punching the Ticket to More Efficient Materials Development"," July 2021.

[81] G. Agarwal, H. A. Doan, L. A. Robertson, L. Zhang, and R. S. Assary, "Discovery of Energy Storage Molecular Materials Using Quantum Chemistry-Guided Multiobjective Bayesian Optimization," *Chemistry of Materials*, vol. 33, pp. 8133–8144, Oct. 2021.

[82] C. Kim, T. Doan Huan, S. Krishnan, and R. Ramprasad, "A hybrid organic-inorganic perovskite dataset," *Scientific Data*, vol. 4, no. 170057, pp. 1–11, 2017.

[83] E. Soedarmadji, H. S. Stein, S. K. Suram, D. Guevarra, and J. M. Gregoire, "Tracking materials science data lineage to manage millions of materials experiments and analyses," *npj Computational Materials*, vol. 5, pp. 1–9, July 2019.

[84] H. S. Stein, D. Guevarra, A. Shinde, R. J. R. Jones, J. M. Gregoire, and J. A. Haber, "Functional mapping reveals mechanistic clusters for OER catalysis across (Cu–Mn–Ta–Co–Sn–Fe)Ox composition and pH space," *Materials Horizons*, vol. 6, pp. 1251–1258, July 2019.

[85] D. Bash, Y. Cai, V. Chellappan, S. L. Wong, X. Yang, P. Kumar, J. D. Tan, A. Abutaha, J. J. Cheng, Y.-F. Lim, *et al.*, "Multi-fidelity high-throughput optimization of electrical conductivity in p3ht-cnt composites," *Advanced Functional Materials*, p. 2102606, 2021.

[86] F. Mekki-Berrada, Z. Ren, T. Huang, W. K. Wong, F. Zheng, J. Xie, I. P. S. Tian, S. Jayavelu, Z. Mahfoud, D. Bash, *et al.*, "Two-step machine learning enables optimized nanoparticle synthesis," *npj Computational Materials*, vol. 7, no. 1, pp. 1–10, 2021.

[87] S. Sun, A. Tiihonen, F. Oviedo, Z. Liu, J. Thapa, Y. Zhao, N. T. P. Hartono, A. Goyal, T. Heumueller, C. Batali, *et al.*, "A data fusion approach to optimize compositional stability of halide perovskites," *Matter*, vol. 4, no. 4, pp. 1305–1322, 2021.

[88] J. R. Deneault, J. Chang, J. Myung, D. Hooper, A. Armstrong, M. Pitt, and B. Maruyama, "Toward autonomous additive manufacturing: Bayesian optimization on a 3d printer," *MRS Bulletin*, pp. 1–10, 2021.

[89] A.-C. Bédard, A. Adamo, K. C. Aroh, M. G. Russell, A. A. Bedermann, J. Torosian, B. Yue, K. F. Jensen, and T. F. Jamison, "Reconfigurable system for automated optimization of diverse chemical reactions," *Science*, vol. 361, pp. 1220–1225, Sept. 2018.

[90] D. T. Ahneman, J. G. Estrada, S. Lin, S. D. Dreher, and A. G. Doyle, "Predicting reaction performance in C–N cross-coupling using machine learning," *Science*, vol. 360, no. 6385, 2018.

[91] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[92] M. J. Anderson and P. J. Whitcomb, *DOE simplified: practical tools for effective experimentation.* CRC Press, 2016.

[93] G. E. P. Box, J. S. Hunter, and W. G. Hunter, *Statistics for experimenters: design, innovation and discovery*, vol. 2. 2005.

[94] R. A. Fisher, *The design of experiments.* Oliver and Boyd; Edinburgh; London, 1937.

[95] H. B. Curry, "The method of steepest descent for nonlinear minimization problems," *Quarterly of Applied Mathematics*, vol. 2, no. 3, pp. 258–261, 1944.

[96] H. Bouwmeester, A. Dougherty, and A. V. Knyazev, "Nonsymmetric preconditioning for conjugate gradient and steepest descent methods," *Procedia Computer Science*, vol. 51, pp. 276–285, 2015.

[97] M. R. Hestenes, E. Stiefel, *et al.*, "Methods of conjugate gradients for solving linear systems," *Journal of research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[98] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[99] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on scientific computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[100] J. Nocedal and S. J. Wright, "Sequential quadratic programming," *Numerical optimization*, pp. 529–562, 2006.

[101] D. Kraft *et al.*, "A software package for sequential quadratic programming," 1988.

[102] W. Huyer and A. Neumaier, "Snobfit–stable noisy optimization by branch and fit," *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 2, pp. 1–25, 2008.

[103] D. J. Wales and J. P. Doye, "Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms," *The Journal of Physical Chemistry A*, vol. 101, no. 28, pp. 5111–5116, 1997.

[104] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[105] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "Deap: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, 2012.

[106] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[107] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[108] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, Ieee, 1995.

[109] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, pp. 69–73, IEEE, 1998.

[110] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[111] The GPyOpt authors, "Gpyopt: A bayesian optimization framework in python." `http://github.com/SheffieldML/GPyOpt`, 2016.

[112] J. S. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," *JMLR*, 2013.

[113] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, pp. 2546–2554, 2011.

[114] J. S. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[115] F. Häse, L. M. Roch, C. Kreisbeck, and A. Aspuru-Guzik, "Phoenics: A bayesian optimizer for chemistry," *ACS central science*, vol. 4, no. 9, pp. 1134–1145, 2018.

[116] M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, T. Ruhkopf, R. Sass, and F. Hutter, "Smac3: A versatile bayesian optimization package for hyperparameter optimization," 2021.

[117] K. Kandasamy, K. R. Vysyaraju, W. Neiswanger, B. Paria, C. R. Collins, J. Schneider, B. Poczos, and E. P. Xing, "Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly.," *J. Mach. Learn. Res.*, vol. 21, no. 81, pp. 1–27, 2020.

[118] M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy, "Botorch: A framework for efficient monte-carlo bayesian optimization," 2020.

[119] A. I. Cowen-Rivers, W. Lyu, R. Tutunov, Z. Wang, A. Grosnit, R. R. Griffiths, H. Jianye, J. Wang, and H. B. Ammar, "An empirical study of assumptions in bayesian optimisation," *arXiv preprint arXiv:2012.03826*, 2020.

[120] F.-M. De Rainville, F.-A. Fortin, M.-A. Gardner, M. Parizeau, and C. Gagné, "Deap: A python framework for evolutionary algorithms," in *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pp. 85–92, 2012.

[121] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: a python library for model selection and hyperparameter optimization," *Computational Science & Discovery*, vol. 8, no. 1, p. 014008, 2015.

[122] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration," in *Advances in Neural Information Processing Systems*, 2018.

[123] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen, and R. Barzilay, "Analyzing Learned Molecular Representations for Property Prediction," *Journal of Chemical Information and Modeling*, vol. 59, pp. 3370–3388, Aug. 2019. Publisher: American Chemical Society.

[124] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, V. M. Tran, A. Chiappino-Pepe, A. H. Badran, I. W. Andrews, E. J. Chory, G. M. Church, E. D. Brown, T. S. Jaakkola, R. Barzilay, and J. J. Collins, "A Deep Learning Approach to Antibiotic Discovery," *Cell*, vol. 180, pp. 688–702.e13, Feb. 2020. Publisher: Elsevier.

[125] E. Heid and W. H. Green, "Machine Learning of Reaction Properties via Learned Representations of the Condensed Graph of Reaction," *Journal of Chemical Information and Modeling*, vol. 62, pp. 2101–2110, May 2022. Publisher: American Chemical Society.

[126] The Chemprop authors, "Chemprop." `http://chemprop.csail.mit.edu/`, 2019.

[127] D. Perera, J. W. Tucker, S. Brahmbhatt, C. J. Helal, A. Chong, W. Farrell, P. Richardson, and N. W. Sach, "A platform for automated nanomole-scale reaction screening and micromole-scale synthesis in flow," *Science*, vol. 359, no. 6374, pp. 429–434, 2018.

[128] M. Ponce, R. van Zon, S. Northrup, D. Gruner, J. Chen, F. Ertinaz, A. Fedoseev, L. Groer, F. Mao, B. C. Mundim, *et al.*, "Deploying a top-100 supercomputer for large parallel workloads: The niagara supercomputer," in *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*, pp. 1–8, 2019.

[129] C. Loken, D. Gruner, L. Groer, R. Peltier, N. Bunn, M. Craig, T. Henriques, J. Dempsey, C.-H. Yu, J. Chen, *et al.*, "Scinet: lessons learned from building a power-efficient top-20 system and data centre," in *Journal of Physics-Conference Series*, vol. 256, p. 012026, 2010.

[130] R. J. Hickman, M. Aldeghi, F. Häse, and A. Aspuru-Guzik, "Bayesian optimization with known experimental and design constraints for chemistry applications," 2022.

[131] B. J. Reizman, Y.-M. Wang, S. L. Buchwald, and K. F. Jensen, "Suzuki–miyaura cross-coupling optimization enabled by automated feedback," *Reaction chemistry & engineering*, vol. 1, no. 6, pp. 658–666, 2016.

[132] J. Cheng, "A neural network approach to ordinal regression," *CoRR*, vol. abs/0704.1028, 2007.

[133] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, Dec. 1998.

Supplementary Information

Olympus, enhanced: benchmarking mixed-parameter and multi-objective optimization in chemistry and materials science

Riley J. Hickman,[1,2,3*] Priyansh Parakh,[1,4] Austin Cheng,[1,2,3] Qianxiang Ai,[5] Joshua Schrier,[5] Matteo Aldeghi,[6] and Alán Aspuru-Guzik[1,2,3,7,8,9†]

[1] *Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, ON M5S 3H6, Canada*
[2] *Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4, Canada*
[3] *Vector Institute for Artificial Intelligence, Toronto, ON M5S 1M1, Canada*
[4] *Faculty of Applied Science and Engineering, University of Toronto, ON M5S 1A4, Canada*
[5] *Department of Chemistry, Fordham University, 441 E. Fordham Road, The Bronx, New York 10458, United States*
[6] *Google Research, Mountain View, CA 94043, United States*
[7] *Department of Chemical Engineering & Applied Chemistry, University of Toronto, Toronto, ON M5S 3E5, Canada*
[8] *Department of Materials Science & Engineering, University of Toronto, Toronto, ON M5S 3E4, Canada*
[9] *Lebovic Fellow, Canadian Institute for Advanced Research, Toronto, ON M5G 1Z8, Canada*
[*] *riley.hickman@mail.utoronto.com*
[†] *alan@aspuru.com*

S.1. LIST OF NEW DATASETS

A. `diffvap_crystal`

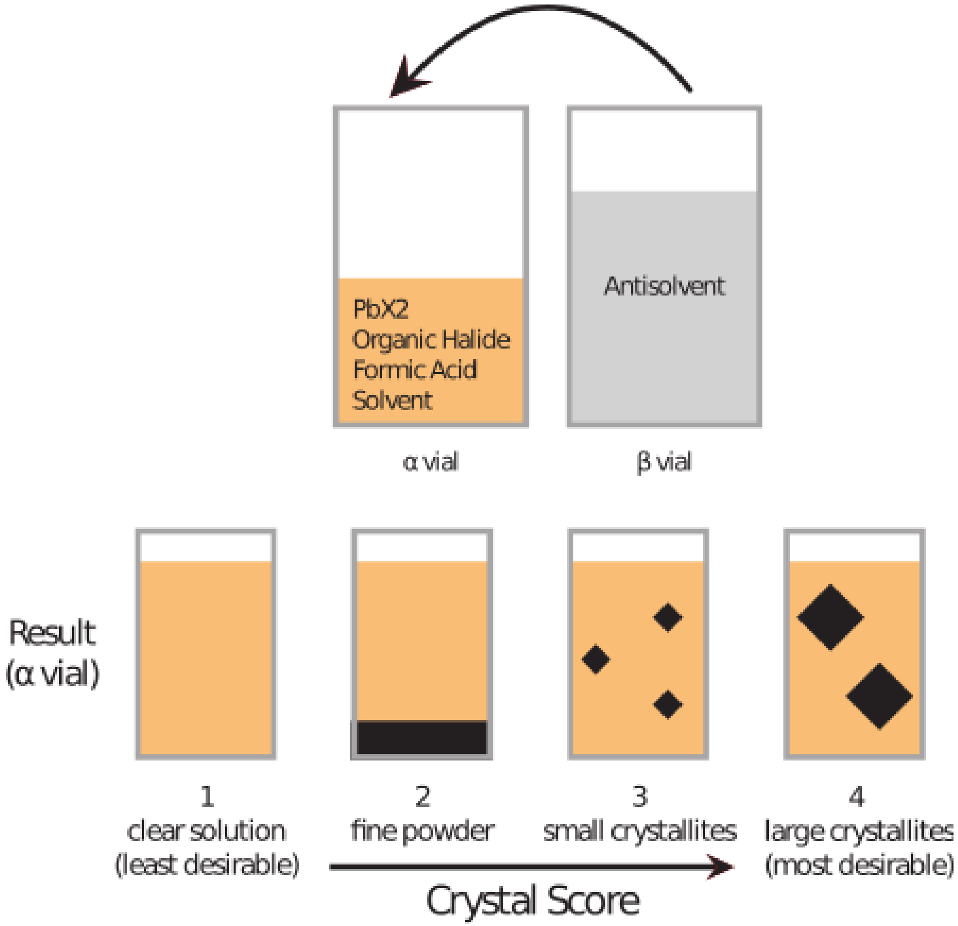


FIG. S1. Experiment setup and result classification for vapor diffusion crystallization.

The diffvap_crystal dataset reports the results for high-throughput antisolvent vapor diffusion crystallization of metal halide perovskitoids. This dataset contains 918 experimental measurements previously described in Refs.[19,78] Briefly, two glass vials were used for each reaction. The α vial, which contained a solution mixture of organoammonium halide salt, inorganic halide (lead iodide or bromide), solvent, and formic acid, and the β vial, which contained the volatile antisolvent (dichloromethane), were placed in a sealed chamber at an elevated temperature to allow vapor diffusion from the β vial to the α vial. The experimental setup is depicted in Fig. S1. The experimental outcome is a qualitative description of crystal growth. The crystal_score objective has 4 options (ordered from least to most desirable): clear solution (no solid), fine powder, small crystallites, and large (> 0.1 mm) crystallites. We train an ordinal regression emulator on this dataset (see SI Sec. S.3 for further details).

Technically, the diffvap_crystal dataset features *a priori* known constraints on the reagent concentration space and the nature of the automated experiments. The α vial solution results from mixing a stock solution of organic halide, a stock solution of organic halide *and* lead halide, formic acid, and the required amount of solvent to reach the desired concentrations. Thus the composition of the α vial is constrained to the convex hull defined by the stock solution composition, which are in turn limited by the solubility of the lead halide and organoammonium halide salts. Although optimization over constrained domains is within the ability of several experiment planners in Olympus,[130] we do not provide a known constraints module here, and reserve its implementation for future work. Instead, we assign infeasible compositions (the compositions cannot be reached using stock solutions) with the least desirable crystal_score, clear solution.

TABLE S3. Parameter space for the diffvap_crystal dataset. All concentrations and volumes were measured at the beginning of the reaction.

Parameter	Kind	Range/num options	Description	Descriptors
organic	categorical	17	organic halide identity	Yes
organic molarity	continuous	$[0.01434 - 7.3935]$	concentration of organic halide (mol/L)	N/A
solvent	categorical	3	solvent identity	Yes
solvent molarity	continuous	$[1.05569 - 12.79558]$	concentration of solvent (mol/L)	N/A
inorganic molarity	continuous	$[0.0 - 2.26115]$	concentration of inogranic halide (mol/L)	N/A
acid molarity	continuous	$[0.0 - 22.42276]$	concentration of formic acid (mol/L)	N/A
alpha vial volume	continuous	$[0.000149, 0.000744]$	volume of α vial (L)	N/A
beta vial volume	continuous	$[0.001, 0.0008]$	volume of β vial (L)	N/A
reaction time	discrete	3	vapor diffusion time (second)	N/A
reaction temperature	discrete	3	vapor diffusion temperature (Celsius)	N/A

TABLE S4. Objectives for the diffvap_crystal dataset. Note that the objective here is an ordinal variable.

Objective	Description	Goal
crystal score	qualitative description of perovskitoid crystallization	maximize

B. dye_lasers

The dye_lasers dataset reports computed photophysical properties for 3458 organic molecules synthesized from three groups of molecular building blocks – A, B, and C (resulting in A-B-C-B-A pentamers).[79] Three syntheses are used: iterative Suzuki-Miyaura cross-coupling reactions, nucleophilic aromatic substitutions and Buchwald-Hartwig aminations (depicted in Figure S2). Each molecule was subjected to a computational protocol consisting of cheminformatic, semi-empirical and *ab initio* quantum chemical steps to compute absorption and emission spectra and fluorescence rates. The objectives of this dataset, in order of decreasing importance are i) the peak score, which is a dimensionless quantity given by the fraction of the fluorescence power spectral density that falls within the $400 - 460$ nm region, ii) the spectral overlap of the absorption and emission spectra, and iii) the fluorescence rate.

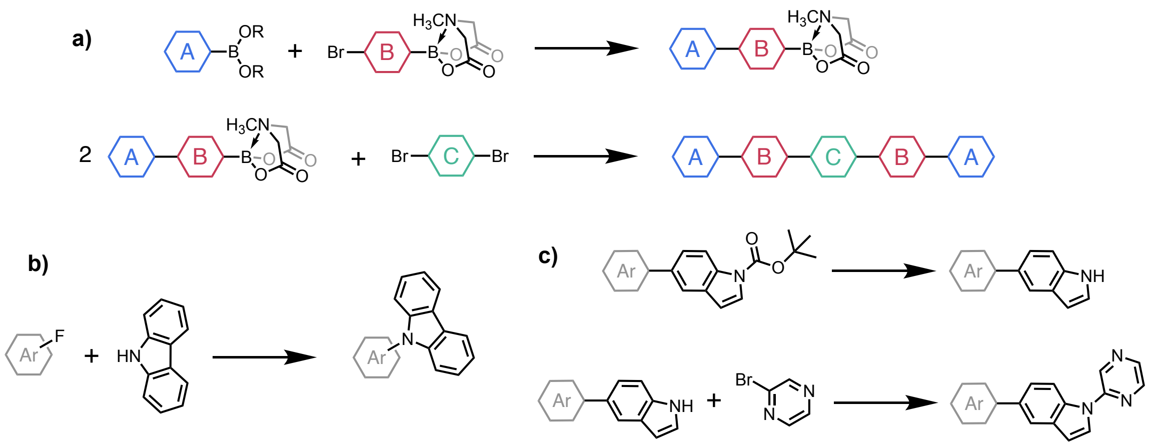


FIG. S2. The three syntheses used for the symmetric A-B-C-B-A pentamers in the dye_lasers dataset. a) iterative Suzuki-Miyaura cross-coupling reaction, b) nucleophilic aromatic substitution, c) Buchwald-Hartwig amination.

TABLE S5. Parameter space for the dye_lasers dataset.

Parameter	**Kind**	**Range / num options**	**Description**	**Descriptors**
A fragment	categorical	14	terminal fragment	None
B fragment	categorical	13	bridge fragment	None
C fragment	categorical	19	core fragment	None

TABLE S6. Objectives for the dye_lasers dataset.

Objective	**Description**	**Goal**
Peak score	fraction of fluorescence spectra in 400−460 nm region [a.u.]	maximize
Spectral overlap	overlap of absorption and emission spectra [a.u.]	minimize
Fluorescence rate	fluorescence rate constant [ns^{-1}]	maximize

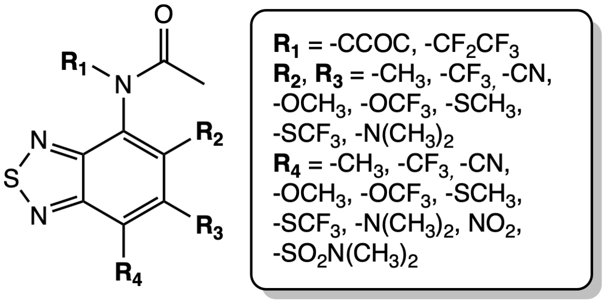

O
R1
N
R2
R3
R4
N
S
N
R1 = -CCOC, -CF2CF3
R2, R3 = -CH3, -CF3, -CN, -OCH3, -OCF3, -SCH3, -SCF3, -N(CH3)2
R4 = -CH3, -CF3, -CN, -OCH3, -OCF3, -SCH3, -SCF3, -N(CH3)2, NO2, -SO2N(CH3)2

FIG. S3. Markush structure of the benzothiadiazole scaffold, along with all the potential substituents in the redoxmers dataset. The entire design space consists of 1408 candidates (2 R_1 × 8 R_2 × 8 R_3 × 11 R_4 options).

C. redoxmers

The redoxmers dataset reports maximum absorption wavelengths, reduction potentials against a Li/Li$^+$ reference electrode, and solvation free energies computed using DFT for a dataset of 1408 benzothiadiazole derivatives.[81] The molecules in this dataset are screened as candidates for self-reporting redox-active materials for non-aqeuous redox flow batteries. The Markush structure of the benzothiadiazole scaffold, along with all the potential substituents are shown in Figure S3. We provide simple physicochemical descriptors for each of the substituents.

TABLE S7. Parameter space for the `redoxmers` dataset.

Parameter	**Kind**	**Range / num options**	**Description**	**Descriptors**
R_1 substituent	categorical	2	substituted group at R_1	Yes
R_2 substituent	categorical	8	substituted group at R_2	Yes
R_3 substituent	categorical	8	substituted group at R_3	Yes
R_4 substituent	categorical	11	substituted group at R_4	Yes

TABLE S8. Objectives for the `redoxmers` dataset.

Objective	**Description**	**Goal**
max absorption difference	absolute difference between λ^{max} and 375 nm [nm]	minimize
reduction potential	reduction potential against $\mathrm{Li/Li^+}$ reference electrode [V]	minimize
solvation free energy	computed solvation free energy (acetonitrile solvent) [eV]	minimize

D. `perovskites`

The `perovskites` dataset reports simulated bandgaps (HSE06 level of theory) for 192 hybrid organic-inorganic perovskite (HOIP) materials.[82] The HOIP candidates of this dataset are designed from a set of 4 different halide anions, 3 different group-IV cations and 16 different organic anions. Electronic and geometric descriptors of the HOIP components are also provided. We characterize the inorganic constituents (anion and cation) by their electron affinity, ionization energy, mass, and electronegativity. Organic components are described by their HOMO and LUMO energies, dipole moment, atomization energy, radius of gyration, and molecular weight.

TABLE S9. Parameter space for the `perovskites` dataset.

Parameter	**Kind**	**Range / num options**	**Description**	**Descriptors**
organic	categorical	16	organic anion	Yes
cation	categorical	3	group-IV cation	Yes
anion	categorical	4	halide anion	Yes

TABLE S10. Objectives for the `perovskites` dataset.

Objective	**Description**	**Goal**
HSE06 gap	bandgap from HSE06 level of theory [eV]	minimize

E. `oer_plate_a` – `oer_plate_d`

The `oer_plate` datasets comprise 4 high-thoughput screens for oxygen evolution reaction (OER) activity by systematically exploring high-dimensional chemical spaces.[25,83,84] The four datasets each contain a discrete library of 2121 catalysts, comprising all unary, binary, ternary and quaternary compositions from unique 6 element sets with 10 at% intervals. The composition systems for each of the four datasets are as follows, a) Mn-Fe-Co-Ni-La-Ce, b) Mn-Fe-Co-Ni-Cu-Ta, c) Mn-Fe-Co-Cu-Sn-Ta, and d) Ca-Mn-Co-Ni-Sn-Sb. During the optimizations, experiment planning strategies traverse the entire standard 6-simplex of catalyst compositions. Stein *et al.*[84] report only unary, binary, ternary and quaternary compositions from 6 element sets with 10 at% intervals. For the compositions whose overpotentials are not reported in the original dataset, *i.e.*, the quinary and senary compositions, a probabilistic emulator is used to produce a virtual measurement. We do not claim that these extrapolated values are quantitatively accurate with respect to experiment, only that they are reasonable values for overpotentials with respect to measured values.

The oer_plate datasets feature an additional constraint that valid parameters must be on the 6-simplex. To enforce this, we allow the experiment planner to operate on the standard 5-cube, and map proposals to the 6-simplex using a deterministic transformation before they are processed by the emulator.

TABLE S11. Parameter space for the oer_plate datasets.

Parameter	**Kind**	**Range / num options**	**Description**	**Descriptors**
mat 1	continuous	[0.0 – 1.0]	fractional composition of system material 1 [a.u.]	N/A
mat 2	continuous	[0.0 – 1.0]	fractional composition of system material 2 [a.u.]	N/A
mat 3	continuous	[0.0 – 1.0]	fractional composition of system material 3 [a.u.]	N/A
mat 4	continuous	[0.0 – 1.0]	fractional composition of system material 4 [a.u.]	N/A
mat 5	continuous	[0.0 – 1.0]	fractional composition of system material 5 [a.u.]	N/A
mat 6	continuous	[0.0 – 1.0]	fractional composition of system material 6 [a.u.]	N/A

TABLE S12. Objectives for the oer_plate datasets.

Objective	**Description**	**Goal**
overpotential	OER overpotential [V]	minimize

F. p3ht

The p3ht dataset reports the electrical conductivity of composite thin films prepared using a machine learning-driven automated flow mixing setup with a high-throughput drop-casting system.[85] Regio-regular poly-3-hexylthiophene (rr-P3HT) is combined with 4 types of carbon nanotubes (CNTs), leading to to different morphologies and crystaline structures which modulate the electrical conductivity of the thin film. The types of CNTs used in the study are i) long single wall CNTs (*l*-SWNTs, 5-30 μm), ii) short single wall CNTs (*s*-SWNTs, 1-3 μm), iii) multi-walled CNTs (MWCNTs), and iv) double-walled CNTs (MWCNTs). The films are processed by optical and electrical diagnostics to asses their electrical conductivity, which is meant to be maximized.

TABLE S13. Parameter space for the p3ht dataset.

Parameter	**Kind**	**Range / num options**	**Description**	**Descriptors**
p3ht content	continuous	[15.0 – 96.27]	rr-P3HT polymer content [a.u.]	N/A
d1 content	continuous	[0.0 – 60.0]	*l*-SWNT carbon nanotube content [a.u.]	N/A
d2 content	continuous	[0.0 – 70.0]	*s*-SWNT carbon nanotube content [a.u.]	N/A
d6 content	continuous	[0.0 – 85.0]	MWCNT carbon nanotube content [a.u.]	N/A
d8 content	continuous	[0.0 – 75.0]	DWCNT carbon nanotube content [a.u.]	N/A

TABLE S14. Objectives for the p3ht dataset.

Objective	**Description**	**Goal**
conductivity	electrical conductivity [S/cm]	maximize

G. agnp

The agnp dataset is the result of an optimization of silver nanoparticles (AgNPs) for targeted absorbance spectra using a machine learning-driven high-throughput microfluidic platform.[86] The AgNP synthesis was carried out with a droplet-based platform with 5 continuous-valued parameters. Four of the parameters ($Q_{\mathrm{AgNO_3}}$, Q_{PVA}, Q_{TSC}, and Q_{seed}) are flow rate ratios, where Q_i is the ratio between the flow rate of reactant i and the total aqueous flow rate. The fifth parameter, Q_{total}, is the total flow rate. The objective of the optimization is the theoretical absorbance spectrum of triangular prism AgNPs with 50 nm edges and 10 nm heights as calculated by plasmon resonance simulation using discrete dipole scattering. The resulting value is termed the "spectrum score", whose value is to be maximized.

TABLE S15. Parameter space for the `agnp` dataset.

Parameter	Kind	Range / num options	Description	Descriptors
$Q_{\mathrm{AgNO_3}}$	continuous	[4.53 – 42.80981595]	silver nitrate flow rate ration [%]	N/A
Q_{PVA}	continuous	[9.999518 – 40.00101474]	polyvinyl alcohol flow rate ratio [%]	N/A
Q_{TSC}	continuous	[0.5 – 30.5]	trisodium citrate flow rate ratio [%]	N/A
Q_{seed}	continuous	[0.498851653 – 19.5]	silver seed flow rate ratio [%]	N/A
Q_{total}	continuous	[200.0 – 983.0]	total (oil and aqueous phases) flow rate [μL/min]	N/A

TABLE S16. Objectives for the `agnp` dataset.

Objective	Description	Goal
spectrum score	similarity of experimental spectra to theoretical spectra [a.u.]	maximize

H. `thin_films`

The `thin_films` dataset reports the results of a closed-loop machine-learning driven optimization of the stability of lead iodide perovskite materials that suffer from heat- and moisture-induced degradation.[87] The material search space is the five-element space $\mathrm{Cs}_x\mathrm{MA}_y\mathrm{FA}_{1-x-y}\mathrm{PbI}_3$. Thin-film samples are spin-coated before being examined under 85% relative humidity and 85°C. The objective of this dataset is minimization of the perovskite material's instability index, which is defined as the integrated color change of the films over the accelerated degradation period.

TABLE S17. Parameter space for the `thin_films` dataset.

Parameter	Kind	Range / num options	Description	Descriptors
CsPbI	continuous	[0.0 – 1.0]	fractional composition of CsPbI	N/A
FAPbI	continuous	[0.0 – 1.0]	fractional composition of FAPbI	N/A
MAPbI	continuous	[0.0 – 1.0]	fractional composition of MAPbI	N/A

TABLE S18. Objectives for the `thin_films` dataset.

Objective	Description	Goal
instability index	integrated color change over degradation period [a.u.]	minimize

I. `crossed_barrel`

The `crossed_barrel` dataset reports the results from a data-driven optimization of 3D printed parts for their mechanical properties (in this case, their toughness).[27] A platform which combines additive manufacturing, robotics, and mechanical testing was employed. The system prints a crossed barrel family of structures, which are supported by n hollow columns with outer radius r and thickness t, twisted at an angle θ. After printing, the structures are subjected to uni-axial compression. The toughness objective is then recorded as the area under the resulting force-displacement curve, and is intended to be maximized.

TABLE S19. Parameter space for the `crossed_barrel` dataset.

Parameter	Kind	Range / num options	Description	Descriptors
n	discrete	$6 - 12$	number of hollow columns	N/A
θ	continuous	$[0.0 - 200.0]$	twist angle of the columns [degrees]	N/A
r	continuous	$[1.5 - 2.5]$	outer radius of the columns [mm]	N/A
t	continuous	$[0.7 - 1.4]$	thickness of the hollow columns [mm]	N/A

TABLE S20. Objectives for the `crossed_barrel` dataset.

Objective	Description	Goal
toughness	mechanical toughness [J]	maximize

J. `autoam`

The `autoam` dataset reports the result of an autonomous optimization of four continuous-valued 3d printing parameters to optimize the geometry of the leading segment of printed lines to target specifications.[88] The objective is termed the "shape score", which measures the similarity between the printed line and the target specifications, and should be maximized.

TABLE S21. Parameter space for the `autoam` dataset.

Parameter	Kind	Range / num options	Description	Descriptors
prime delay	continuous	$[0.0 - 5.0]$	delay before deposition commencement [s]	N/A
print speed	continuous	$[0.1 - 10.0]$	deposition rate [mm s^{-1}]	N/A
x offset correction	continuous	$[-1.0 - 1.0]$	x-component of offset vector [mm]	N/A
y offset correction	continuous	$[-1.0 - 1.0]$	y-component of offset vector [mm]	N/A

TABLE S22. Objectives for the `autoam` dataset.

Objective	Description	Goal
shape score	similarity between printed line and target specifications [a.u.]	maximize

K. suzuki_i – suzuki_iv

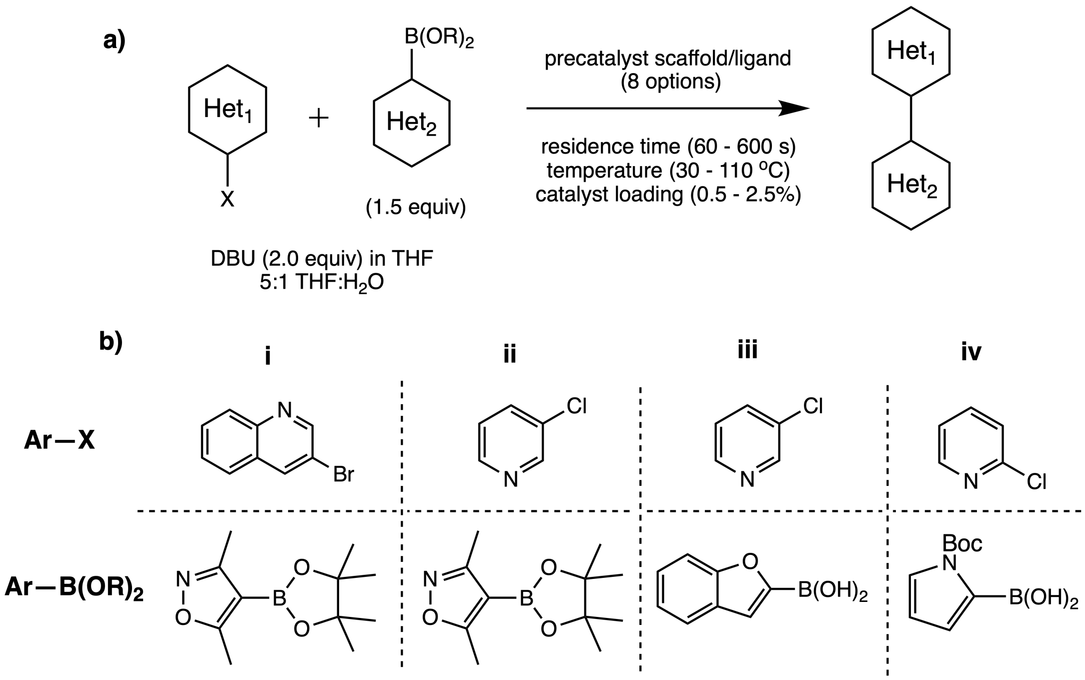


FIG. S4. a) General scheme for Suzuki–Miyaura cross-coupling of two heterocycles in the presence of 1,8-diazabicyclo[5.4.0]undec-7-ene (DBU) and THF/water. b) Structure of the substrates in each of the four Suzuki-Miyaura case studies (corresponding to suzuki_i - suzuki_iv datasets).

The datasets suzuki_i – suzuki_iv report yeild and catalyst turnover number for flow-based Suzuki-Miyaura cross-coupling reactions with varying substrates.[131] There are three continuous parameters (temperature, residence time, and catalyst loading) and one categorical parameter (Pd catalyst ligand). The objective is to simultaneously maximize both the yield and catalyst turnover number.

TABLE S23. Parameter spaces for the suzuki_i – suzuki_iv datasets.

Parameter	Kind	Range / num options	Description	Descriptors
ligand	categorical	8	Pd catalyst ligand	None
res time	continuous	[60.0 – 600.0]	reaction residence time [s]	N/A
temperature	continuous	[30.0 – 110.0]	reaction temperature [°C]	N/A
catalyst loading	continuous	[0.498 – 2.515]	catalyst loading fraction [a.u.]	N/A

TABLE S24. Objectives for the suzuki_i – suzuki_iv datasets.

Objective	Description	Goal
yield	reaction yield [%]	maximize
turnover	catalyst turnover number [prod/cat]	maximize

L. suzuki_edbo

The suzuki_edbo dataset reports yields for a Suzuki-Miyaura coupling reaction preformed on the nanomole scale using a automated flow-based synthesis platform.[14,89] The reaction scheme, along with the search space is presented in Figure S5. This dataset consists of 5 categorical parmeters: the boronic acid derivative electrophile, aryl halide nucleophile, base used in the deprotonation step, Pd catalyst ligand, and solvent. Collectively, there are 3696 unique reactions.

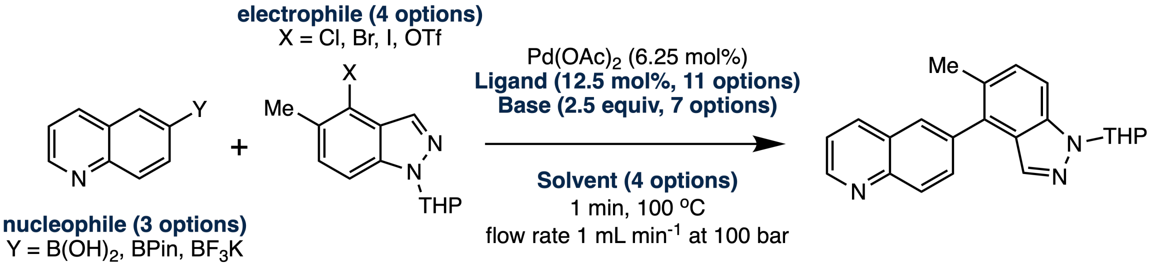


FIG. S5. Reaction scheme for suzuki_edbo dataset.

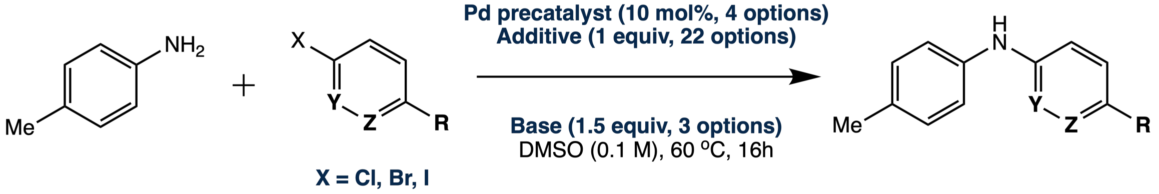
TABLE S25. Parameter space for the suzuki_edbo dataset.

Parameter	Kind	Range / num options	Description	Descriptors
electrophile	categorical	4	boronic acid derivative	Yes
nucleophile	categorical	3	aryl halide	Yes
base	categorical	7	base used in deprotonation step	Yes
ligand	categorical	11	Pd catalyst ligand	Yes
solvent	categorical	4	solvent	Yes

TABLE S26. Objectives for the suzuki_edbo dataset.

Objective	Description	Goal
yield	reaction yield [%]	maximize

M. buchwald_a – buchwald_e


NH_2
Me
+
X
Y
Z
R
X = Cl, Br, I
Pd precatalyst (10 mol%, 4 options)
Additive (1 equiv, 22 options)
Base (1.5 equiv, 3 options)
DMSO (0.1 M), 60 °C, 16h
H
N
Me
Y
Z
R

Reaction	Y	Z	R
buchwald_a	CH	CH	CF_3
buchwald_b	CH	CH	OMe
buchwald_c	CH	CH	Et
buchwald_d	N	CH	H
buchwald_e	CH	N	H

FIG. S6. Reaction scheme for the Buchwald-Hartwig datasets.

The buchwald datasets comprise 5 datasets which each report the yield of Pd-catalyzed Buchwald-Hartwig amination reactions of aryl halides with 4-methylaniline in the presence of varying isoxazole additives, Pd catalyst ligands, and bases obtained via ultra-high-throughput experimentation.[90] Each of the 5 datasets consists of 792 yield measurements. The reaction scheme and parameter space are shown in Fig. S6.

TABLE S27. Parameter space for the Buchwald-Hartwig datasets

Parameter	**Kind**	**Range / num options**	**Description**	**Descriptors**
aryl halide	categorical	3	aryl halide substrate (with Cl, Br or I)	Yes
additive	categorical	22	isoxazole additive	Yes
base	categorical	3	base used in deprotonation step	Yes
ligand	categorical	4	ligand of Pd catalyst	Yes

TABLE S28. Objectives for the Buchwald-Hartwig datasets

Objective	**Description**	**Goal**
yield	yield of Buchwald-Hartwig amination reaction [%]	maximize

S.2. LIST OF NEW SURFACES

A. `cat_camel`

This surface features a degenerate and pseudo-disconnected global minimum. In 2d, it has global minima at $(x_0, x_1) = (7, 11)$ and $(x_0, x_1) = (14, 10)$. The categorical Camel surface is generalized from the Camel function on continuous domains and features a degenerate and pseudo-disconnected global minimum.

B. `cat_michalewicz`

This surface features a sharp well where the global optimum is located. The number of psuedo-local minima scales factorially with the number of dimensions. In 2d, it features a global minima at $(x_0, x_1) = (14, 10)$. The Michalewicz surface is generalized to categorical spaces from the continuous Michalewicz function.

C. `cat_dejong`

The `cat_dejong` surface is inspired by the Dejong function and, as such, represents the generalization of a parabola to categorical spaces. We therefore refer to the Dejong functions as pseudo-convex. Similar to the Ackley surface, the Dejong surface features a well-defined global minimum if the number of options for all dimensions is odd, and a degenerate global minimum if at least one of the dimensions features an even number of options.

D. `cat_slope`

The `cat_slope` surface is constructed such that the response linearly increases with the index of the option along each dimension in the reference ordering. It presents a generalization of a plane to categorical domains.

E. `cat_ackley`

The `cat_ackley` surface is inspired by the Ackley path function for continuous spaces. It features a narrow funnel around the global minimum, which is degenerate if the number of options along one (or more) dimensions is even and well-defined if the number of options for all dimensions is odd.

F. `mult_fonseca`

The `mult_fonseca` function is defined in n parameter dimensions, where $-4 \leq x_i \leq 4 \ \forall \ i \in 1, \ldots, n$ and has two objectives, both of which are to be minimized.

$$f_1(\boldsymbol{x}) = 1 - \exp\left[-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right], \tag{1}$$

$$f_2(\boldsymbol{x}) = 1 - \exp\left[-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right]. \tag{2}$$

G. `mult_viennet`

The `mult_viennet` function is defined in 2 parameter dimensions where $-3 \leq x_1, x_2 \leq 3$ and has three objectives, all of which are to be minimized.

$$f_1(\boldsymbol{x}) = 0.5\left(x_1^2 + x_2^2\right) + \sin\left(x_1^2 + x_2^2\right), \tag{3}$$

$$f_2(\boldsymbol{x}) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15, \tag{4}$$

$$f_3(\boldsymbol{x}) = \frac{1}{(x_1^2 + x_2^2 + 1)} - 1.1e^{-(x_1^2 + x_2^2)}. \tag{5}$$

H. `mult_zdt1`

The `mult_zdt1` surface is the Zitzler–Deb–Thiele's function (N1). This surface is defined between 2 and 30 parameter dimensions where $0 \leq x_i \leq 1 \ \forall \ i \in 1, \ldots, 30$. There are two objectives, both of which are to be minimized.

$$f_1(\boldsymbol{x}) = x_1, \tag{6}$$

$$f_2(\boldsymbol{x}) = g(\boldsymbol{x})h\left(f_1(\boldsymbol{x}), g(\boldsymbol{x})\right), \tag{7}$$

where

$$g(\boldsymbol{x}) = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i, \tag{8}$$

$$h\left(f_1(\boldsymbol{x}), g(\boldsymbol{x})\right) = 1 - \sqrt{\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}}. \tag{9}$$

I. `mult_zdt2`

The `mult_zdt2` surface is the Zitzler–Deb–Thiele's function (N2). This surface is defined between 2 and 30 parameter dimensions where $0 \leq x_i \leq 1 \ \forall \ i \in 1, \ldots, 30$. There are two objectives, both of which are to be minimized.

$$f_1(\boldsymbol{x}) = x_1, \tag{10}$$

$$f_2(\boldsymbol{x}) = g(\boldsymbol{x})h\left(f_1(\boldsymbol{x}), g(\boldsymbol{x})\right), \tag{11}$$

where

$$g(\boldsymbol{x}) = 1 + \frac{9}{29}\sum_{i=2}^{30} x_i, \tag{12}$$

$$h\left(f_1(\boldsymbol{x}), g(\boldsymbol{x})\right) = 1 - \left(\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}\right)^2. \tag{13}$$

J. `mult_zdt3`

The `mult_zdt3` surface is the Zitzler–Deb–Thiele's function (N3). This surface is defined between 2 and 30 parameter dimensions where $0 \leq x_i \leq 1 \; \forall \; i \in 1, \ldots, 30$. There are two objectives, both of which are to be minimized.

$$f_1(\boldsymbol{x}) = x_1 \,, \tag{14}$$

$$f_2(\boldsymbol{x}) = g(\boldsymbol{x}) h\left(f_1(\boldsymbol{x}), g(\boldsymbol{x})\right) \,, \tag{15}$$

where

$$g(\boldsymbol{x}) = 1 + \frac{9}{29} \sum_{i=2}^{30} x_i \,, \tag{16}$$

$$h\left(f_1(\boldsymbol{x}), g(\boldsymbol{x})\right) = 1 - \sqrt{\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}} - \left(\frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})}\right) \sin(10\pi f_1(\boldsymbol{x})) \,. \tag{17}$$

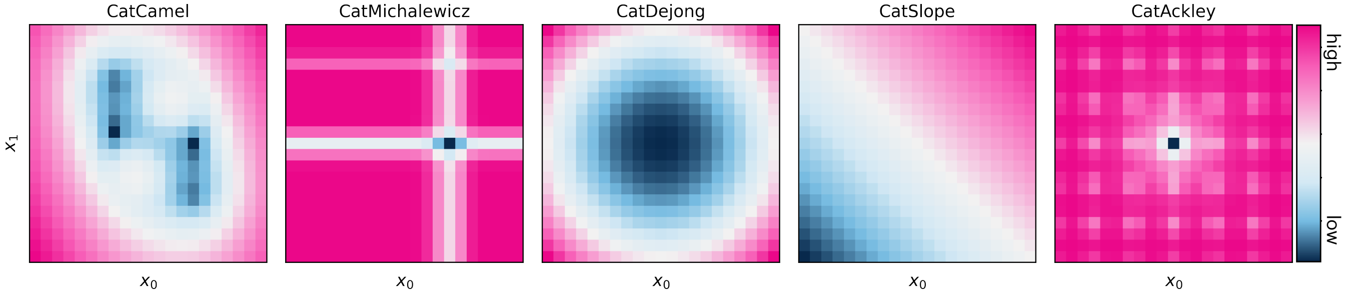


FIG. S7. Visualization of the analytical categorical surfaces included in Olympus. All surfaces are depicted with two parameter dimensions and 21 options per dimension.

S.3. DESCRIPTION AND PERFORMANCE OF NEW EMULATORS

We extended the `Emualtor` module of Olympus to be compatible with multi-output regression. In other words, for the datasets which contain continuous parameters and comprise multiple objectives, we train a single Bayesian neural network (BNN) to estimate all objectives.

We also extend the `Emulator` module to be compatible with optimization problems with ordinal objectives. Ordinal variables are similar to categorical parameters, except that their options have a natural ordering, but the distances between the options are unknown. For example, the `diffvap_crystal` dataset in Olympus targets the optimization of process parameters for a vapour diffusion reaction in which the resulting product is analyzed qualitatively. The `crystal_score` objective has 4 options (ordered from least to most desirable) : `clear_solution`, `fine_powder`, `small_crystallites`, and `large_crystallites`. We use an ordinal regression scheme to train the emulators, following closely the approach reported by Cheng.[132] Ordinal variables with n options are encoded as binary vectors, where, for the m^{th} option, the target vector is $y = (1, 1, 1, \ldots, 0, 0)$, where $y_i = 1$ if $i \leq m$ and 0 otherwise. This encoding is such that the output of our neural network emualtor is related to cumulative probit model for ordinal regression and can be interpreted as a cumulative probability distribution on the n options. The remainder of the training protocol for the BNN emulator remains the same for ordinal regression.

BNN hyperparameters for each dataset are determined using Bayesian optimization via the HyperOpt package.[121] Each optimization proceeds for 50 iterations where the objective is the 5-fold cross-validated RMSD. Parity plots of our emulator model's predictions for single objective datasets are shown in Figure S9. Parity plots for emulator predictions on multi-objecitve datasets are shown in Figure S10. The ordinal regression emulator for the `diffvap_crystal` dataset achieves a train and test set accuracy of 86.9% and 82.1%, respectively.

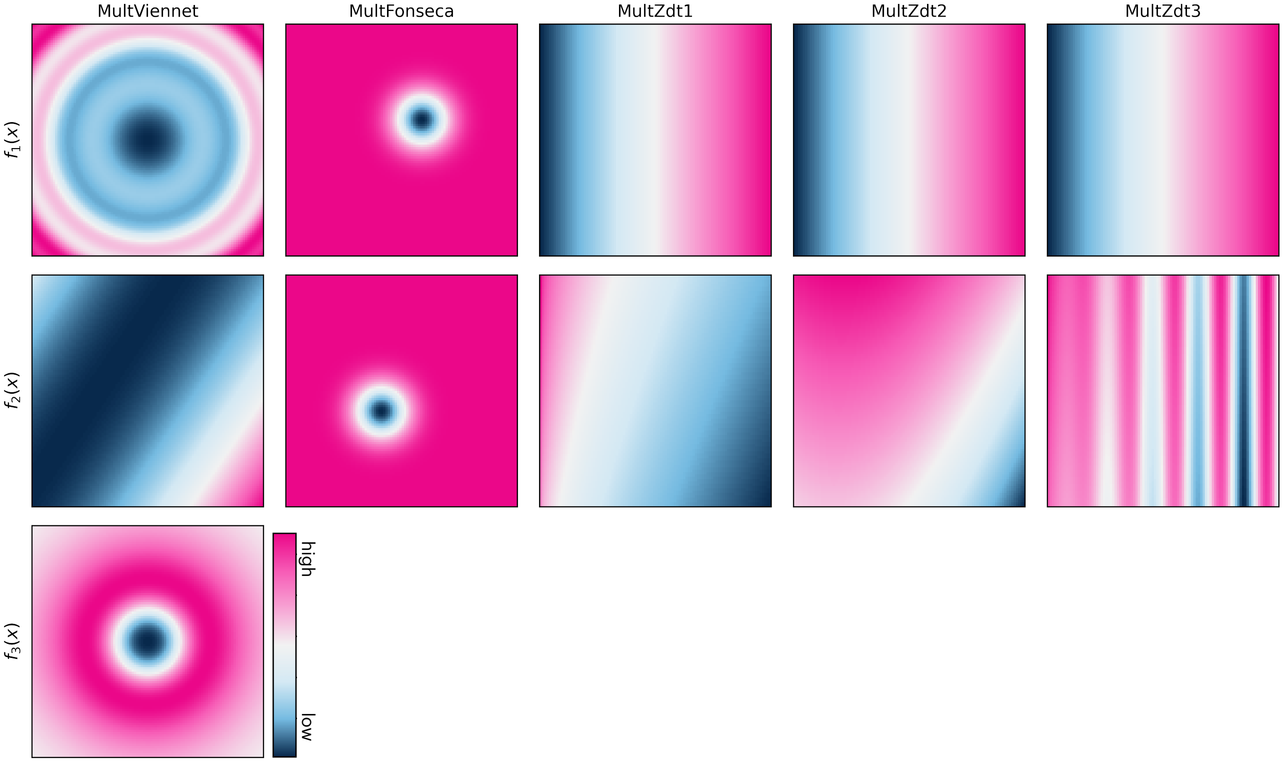


FIG. S8. Visualization of the analytical multi-objective optimization surfaces included in Olympus. All surfaces are depicted with two parameter dimensions.

S.4. ESTIMATING HETEROSCEDASTIC NOISE MODELS FOR FULLY CATEGORICAL DATASETS

Olympus' fully-categorical datasets comprise a discrete set of possible parameter settings for which a corresponding set of measurement(s) is available. We use Bayesian neural networks (BNNs) to capture heteroscedastic aleatoric (data-inherent) uncertainties associated with measurements such that users of Olympus can sample noisy realizations of previous experiments in their optimization benchmarks. The BNN models are trained with variational inference. Variational learning finds the parameters θ of a distribution on BNN weights $q_\theta(\boldsymbol{w})$ which best resembles the true Bayesian posterior on the weights. This procedure provides the predictive distribution for the output y^* given an unseen input instance $\boldsymbol{x}^*$,

$$q^*_\theta\left(y^*|\boldsymbol{x}^*\right) = \int q_\theta(\boldsymbol{w}) p\left(y^*|\mathcal{F}_{\boldsymbol{w}}(\boldsymbol{x}^*)\right) d\boldsymbol{w}\,, \tag{18}$$

where $\mathcal{F}_{\boldsymbol{w}}(\cdot)$ is the output of BNN model $\mathcal{F}$ with weight setting $\boldsymbol{w}$. Under a heteroscedastic model, the the BNN output with configuration $\boldsymbol{w}_t$ is $(\hat{y}^*_t, \hat{\sigma}_t) = \mathcal{F}_{\boldsymbol{w}_t}\left(\boldsymbol{x}^*\right)$, where the t^{th} Monte Carlo weight sample is drawn from $q_\theta(\boldsymbol{w})$. After T such Monte Carlo samples, the total heteroscedastic predictive uncertainty is given by the following expression, which can be divided into contributions from epistemic and aleatoric uncertainties.

$$\widehat{\mathrm{Var}}\left(y^*|\boldsymbol{x}^*\right) = \underbrace{\frac{1}{T}\sum_{t=1}^{T}\left(\hat{y}^*_t\right)^2 - \left(\frac{1}{T}\sum_{t=1}^{T}\hat{y}^*_t\right)^2}_{\text{epistemic}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\hat{\sigma}^2_t}_{\text{aleatoric}}\,. \tag{19}$$

Epistemic uncertainty is related to machine learning model incompleteness, and could in theory be reduced in light of more training observations and/or additional knowledge about the correct model. Aleatoric uncertainty is associated with noise or imprecision on measurements. The latter type better aligns with the kind of uncertainty Olympus seeks to reproduce in optimization benchmark experiments.

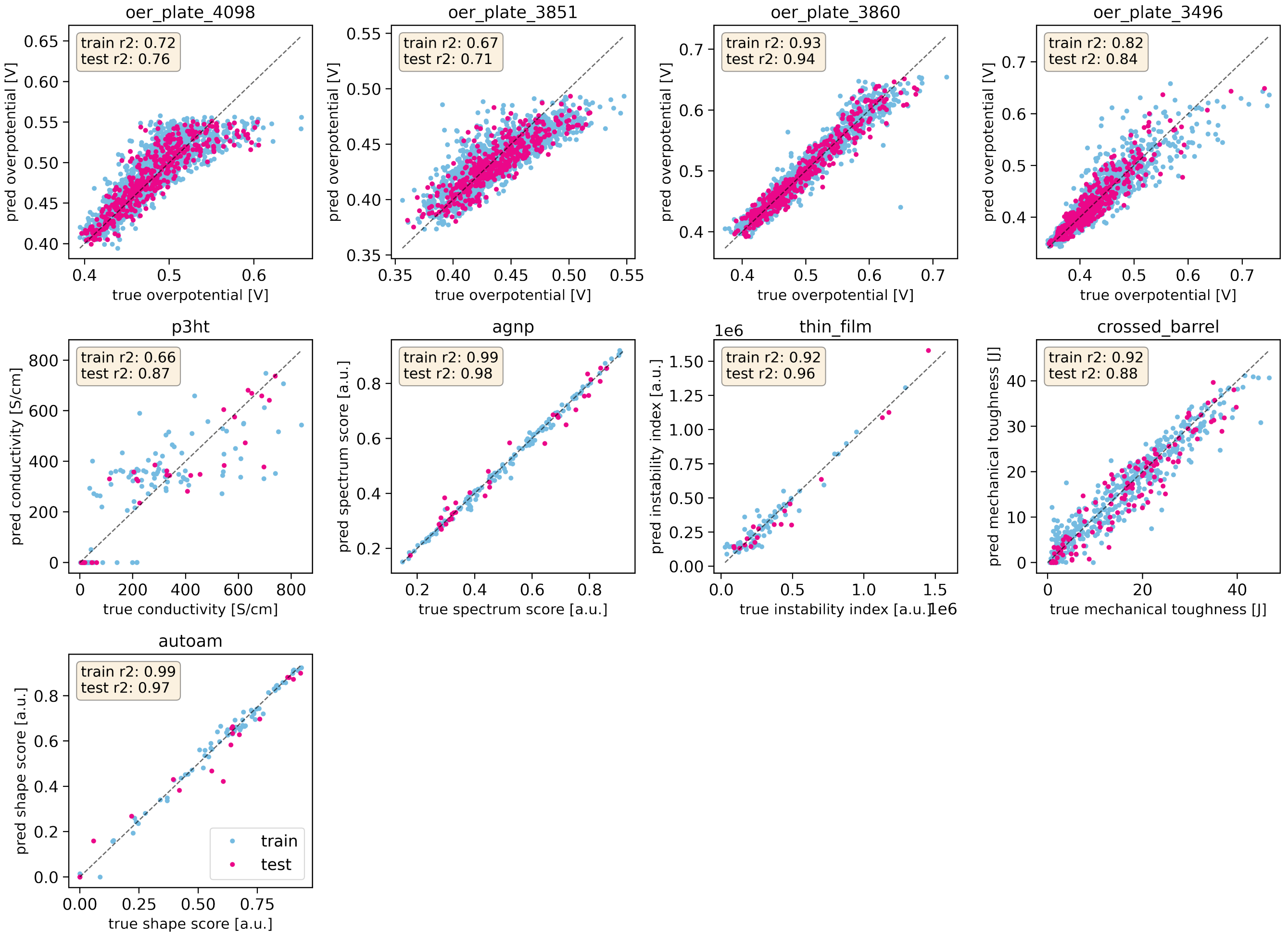


FIG. S9. Emulator parity plots for the new datasets with one single objective.

For a fully-categorical dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, we train a BNN and record its predictive aleatoric variance for each data instance, *i.e.*, for the i^{th} data instance, the aleatoric variance is $\hat{\sigma}_i^2 = \frac{1}{T}\sum_{t=1}^{T}\hat{\sigma}_t^2$ (the last summation on the RHS expression of $\widehat{\mathrm{Var}}\left(y_i^*|\boldsymbol{x}_i^*\right)$). The noisy objective value, $\tilde{y}_i$, returned to the user of Olympus when calling `dataset.run()` for a specific input $\boldsymbol{x}_i$ is then $\tilde{y}_i \sim \mathcal{N}\left(y_i, \hat{\sigma}_i^2\right)$. The discussion thus far considers only scalar-valued objectives, but can be extended to multiple objectives without loss of generality.

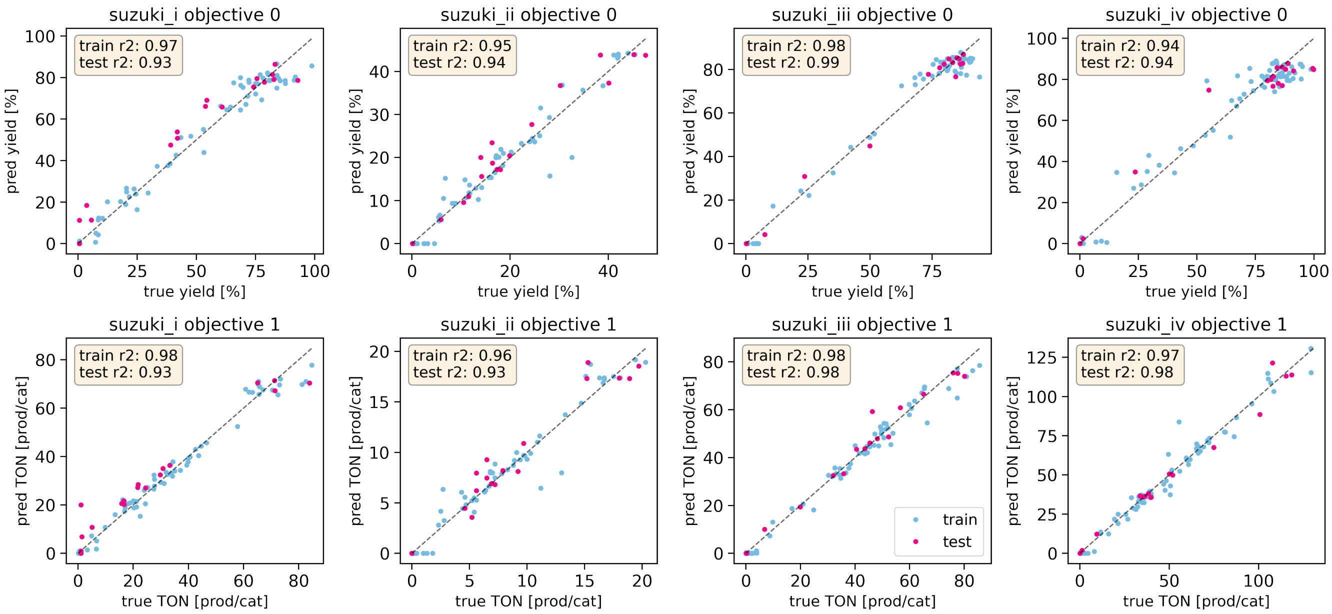


FIG. S10. Emulator parity plots for the `suzuki_i` - `suzuki_iv` datasets.

S.5. DESCRIPTION OF ACHIEVEMENT SCALARIZING FUNCTIONS

In this section, we formally define each of the achievement scalarizing functions (ASFs) used in this work. For the purpose of this discussion, we consider a optimization task over a compact subset of Euclidean space $\mathcal{X} \subset \mathbb{R}^d$. The task features an objective space $\mathcal{Y} \subset \mathbb{R}^n$, corresponding to set of n objective functions, $\boldsymbol{f} = \{f_i\}_{i=1}^n : \mathcal{X} \mapsto \mathcal{Y}$, to be *minimized* concurrently. For parameter setting $\boldsymbol{x} \in \mathbb{R}^d$, noiseless objective values $\boldsymbol{y} \in \mathbb{R}^n$ are obtained as $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$. Furthermore, we assume a dataset of input-output observations $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^K$ has been collected by our optimization procedure.

A. Weighted sum

Weighted sum ASFs maps multiple objectives onto a cumulative scalar objective using a vector of weights $\boldsymbol{w} \in \mathbb{R}^n$ to produce the weighted sum

$$J(\boldsymbol{y}; \boldsymbol{w}) = \sum_{i=1}^{n} w_i y_i \,. \tag{20}$$

Here, we scale each weight w_i by a scaling factor α_i according to the magnitude of the i^{th} objective.

B. Chebyshev

ParEGO[71,72] was introduced to extend the EGO algorithm[133] to a multi-objective optimization setting, but can be used as a general-purpose ASF geared toward expensive-to-evaluate optimization problems. ParEGO is based around the augmented Chebyshev scalarization function. Similar to Weighted Sum, Chebyshev converts n objective values to a single cumulative function via a parameterized scaling weight vector, $\boldsymbol{w}$, whose value is sampled uniformly at each optimization iteration from the set of evenly placed vectors

$$\mathcal{W} = \left\{ \boldsymbol{w} \in \mathbb{R}^n | \sum_{i=1}^{n} w_i = 1 \wedge \forall\, i, w_i = \ell/s, \ell \in \{0, \ldots, s\} \right\} . \tag{21}$$

The parameter s modulates the number of vectors in $\mathcal{W}$, *i.e.*, $|\mathcal{W}| = \binom{s+n-1}{n-1}$. The value of s defaults to 100 in Olympus. The scalar merit of an objective function is computed using the augmented Chebyshev function

$$J(\boldsymbol{y}; \boldsymbol{w}) = \max_{i=1}^{n} (w_i y_i) + \rho \sum_{i=1}^{n} w_i y_i \,, \tag{22}$$

where ρ is a small positive number which defaults to 0.05 in Olympus.

C. Hypervolume indicator

The hypervolume indicator is an example of a set-quality indicator, which facilitate assessment of Pareto fronts by summarizing their characteristics (such as proximity to the Pareto front, diversity and spread) with a single scalar value. Owing to its ease of interpretation, hypervolume is one of the most widely employed set-quality indicators[?] .

The hypervolume indicator maps a set of objective values $\mathcal{D}$ to a measure of the region dominated by that set and bounded above by some reference point $\boldsymbol{r} \in \mathbb{R}^n$. Intuitively, the indicator provides a notion of the size of the covered objective space or the size of the dominated space.[75,77?] Formally, the hypervolume indicator H given a dataset of objective value measurements $\mathcal{D}$ is

$$H(\mathcal{D}; \boldsymbol{r}) = \Lambda\left(\{\boldsymbol{q} \in \mathbb{R}^n \mid \exists \boldsymbol{p} \in \mathcal{D} \,:\, \boldsymbol{p} \leq \boldsymbol{q} \wedge \boldsymbol{q} \leq \boldsymbol{r}\}\right) \,, \tag{23}$$

where Λ is the Lebesgue measure. Here, for two objective space points $\boldsymbol{p} \in \mathbb{R}^n$ and $\boldsymbol{q} \in \mathbb{R}^n$, the expression $\boldsymbol{p} \leq \boldsymbol{q}$ is used to indicate that $\boldsymbol{p}$ *weakly dominates* $\boldsymbol{q}$, that is $p_i \leq q_i \; \forall \; 1 \leq i \leq n$. H can also be described as the union of hyperrectangles

$$H(\mathcal{D}; \boldsymbol{r}) = \Lambda\left(\bigcup_{\boldsymbol{p} \in \mathcal{D},\, \boldsymbol{p} \leq \boldsymbol{r}} [\boldsymbol{p}, \boldsymbol{r}]\right) \,, \tag{24}$$

where $[\boldsymbol{p}, \boldsymbol{r}]$ represents the hyperrectangle fixed from above by reference point $\boldsymbol{r}$ and below by $\boldsymbol{p}$.

In most cases (including this work), the hypervolume indicator is used as an analysis tool to asses the quality and diversity of a Pareto set after an optimization campaign has transpired. However, it can also be used as an ASF to which aims to find solutions which maximize the dominated hypervolume. In this case, the solutions in $\mathcal{D}$ are considered one at a time, which simplifies the calculation of H to the volume of the hyperrectangle with corners at $\boldsymbol{y}$ and $\boldsymbol{r}$,

$$H(\boldsymbol{y}) = \prod_{i=1}^{n} r_i - y_i \,. \tag{25}$$

Importantly, using the hypervolume indicator as a ASF this way produces a constraint that $\boldsymbol{y} \leq \boldsymbol{r}$. As such, we update $\boldsymbol{r}$ at each iteration such that each of its elements correspond to the maximum observed value for that objective in the optimization history $\mathcal{D}$.

D. Chimera

Chimera is an achievement scalarizing function which combines *a priori* scalarizing with lexicographic approaches, and is created for optimization problems where the objective is expensive to evaluate. Chimera allows users to organize multiple objectives into a hierarchy, *i.e.*, $\boldsymbol{f} = \{f_i\}_{i=1}^n$ is replaced by $\boldsymbol{f} = (f_1, \ldots, f_n)$ ordered according to a descending hierarchy of importance (f_1 is more important than f_2, f_2 than f_3, and so on).

User-specified tolerance values $\boldsymbol{y}^{\text{tol}}$ are provided to Chimera which indicate objective value thresholds at which the user is satisfied (e.g., one may want to generate a molecule with a QED of at least 0.6, before prioritizing optimizing its solubility). Whether or not some measured value of the i^{th} objective function y_i satisfies its corresponding tolerance can be indicated by the Heaviside function, $\Theta\left(y_i^{\text{tol}} - y_i\right) = 0$ if $y_i \geq y_i^{\text{tol}}$ and 1 if $y_i < y_i^{tol}$. Alternatively, discontinuities in the cumulative function can be avoided by utilizing a smooth logistic function approximation to Θ, parameterized by smoothing parameter τ,

$$\theta\left(y_i^{\text{tol}} - y_i\right) = \left[1 + \exp\left(-\frac{y_i^{\text{tol}} - y_i}{\tau}\right)\right]^{-1} . \tag{26}$$

Chimera constructs an ASF using approximate Heaviside functions to weight objective functions $\boldsymbol{f}$. Importantly, the resulting ASF is sensitive to only one objective function at a time in any given parameter space region. As such, objective values y_i are shifted based on the minimum of y_{i-1} (the next most important objective function in $\boldsymbol{f}$) in the parameter space regions where y_{i-1} does not satisfy its corresponding tolerance y_i^{tol}. This minimum value is denoted y_{i-1}^{min}. Chimera, $\chi(\mathcal{D}; \boldsymbol{y}^{\text{tol}})$ is then formulated as

$$\chi(\mathcal{D}; \boldsymbol{y}^{\text{tol}}) = y_1 \theta_1^+ + \prod_{i=1}^{n} \left(y_1 - y_{i-1}^{\text{min}}\right) \theta_i^- + \sum_{i=2}^{n} \left(y_i - y_{i-1}^{\text{min}}\right) \theta_i^+ \prod_{j=0}^{i-1} \theta_m^- , \tag{27}$$

where θ_i^+ and θ_i^- are used to abbreviate $\theta\left(y_i^{\text{tol}} - y_i\right)$ and $\theta\left(y_i - y_i^{\text{tol}}\right)$, respectively. We set the smoothing parameter $\tau = 0.001$ in all our experiments.

S.6. PLOTTING AND ANALYSIS

TABLE S29. Supported plot types in Olympus' plotting module. n indicates that the plot type supports an arbitrary number of parameters or objectives. † produces a separate trace subplot for each objective. ‡ produces a scatter plot of all objective measurements with the Pareto optimal values indicated, as well as a visualization of the Pareto front. The `pareto_front` plot supports only *one* optimization campaign with *one* experiment planning strategy at a time.

plot kind	plot type	allowed param types	# allowed objs
`traces`	traces	cont, dis, cat	1
`traces_regret`	traces	cont, dis, cat	1
`traces_rank`	traces	cat	1
`traces_fraction_top_k`	traces	cont, dis, cat	1
`num_evals_top_k`	boxplot	dis, cat	n
`regret_x_evals`	boxplot	cont, dis, cat	n
`hypervolume`	boxplot	cont, dis, cat	≥ 2
`moo_traces` †	traces	cont, dis, cat	≥ 2
`pareto_front` ‡	scatterplot	cont, dis, cat	2–3