Lecture 12

Chemical Design with Bayesian Optimization

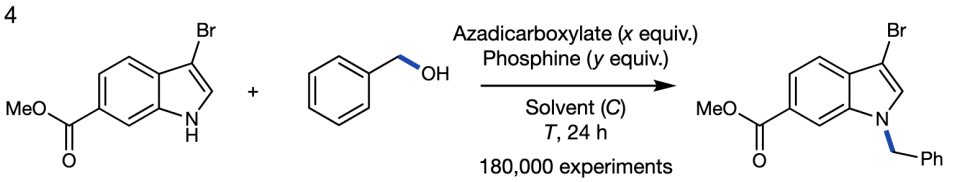
Contents

In the last lecture, we introduced active learning, a framework that guides the acquisition of new data to improve model performance. In this lecture, we turn to a closely related approach, Bayesian optimization (BO), which also sequentially selects new data points. Both frameworks are particularly useful when data acquisition is expensive, as is often the case in chemical experiments or high-level simulations.

Despite this similarity, the objective of Bayesian optimization is fundamentally different from that of active learning. Rather than improving a model globally, BO aims to identify the optimum of a target function, such as reaction yield, reaction rate, or other experimentally measurable quantities. In practice, BO provides a principled strategy for designing chemical experiments or chemical structures that achieve a specific goal.

Bayesian optimization has been widely applied to chemical optimization problems, including reaction optimization [1], chemical synthesis [2], materials design [3], and high-throughput virtual screening (HTVS) [4]. In Section 4, we provide more details of applying BO to these chemical optimization problems.

As a concrete example, we highlight one application in Ref. [2] by the Doyle group, which focuses on optimizing the reaction conditions of the Mitsunobu reaction, where nucleophiles undergo stereospecific coupling with aliphatic alcohols. In this study, Shields et al. applied BO to explore the reaction space for coupling methyl 3-bromo-1H-indole-6-carboxylate with benzyl alcohol. The design variables include (1) 6 azadicarboxylates, (2) 12 phosphines, (3) 5 solvents, as well as continuous parameters such as (4) substrate concentration, (5) azadicarboxylate equivalents, (6) phosphine equivalents, and (7) temperature. Together, these choices define a reaction space of approximately 180,000 possible configurations.



Entry	AD	*x* (equiv.)	Phosphine	*y* (equiv.)	Solvent	*C* (M)	*T* (°C)	Yield
1	DIAD	1.1	PPh_3	1.1	THF	0.1	25	60%
2	ADDP	1.9	$P(Ph)_2Me$	1.9	THF	0.2	25	99%
3	DMAD	1.9	$P(Ph)_2Me$	1.9	MeCN	0.15	45	99%
4	ADDP	1.9	$P(Ph)_2Me$	1.9	THF	0.15	45	99%

Figure 1: Mitsunobu reaction and the high-yield conditions found by Bayesian optimization. [2]

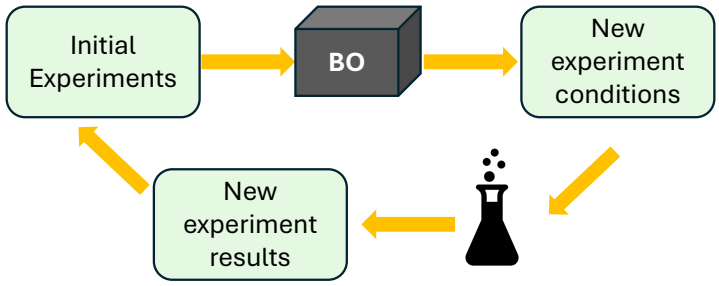


Figure 2: The loop for Bayesian Optimized Mitsunobu reaction.

Using their BO framework [1] , the authors employed (1) DFT-based molecular encoding, (2) a Gaussian process surrogate model, and (3) expected improvement (EI) as the acquisition function (these concepts will be introduced later in the lecture). At each BO iteration, the selected reaction conditions were experimentally evaluated, and the resulting data were added to the training set. Through this iterative process, the authors ultimately identified reaction conditions achieving a 99% yield, substantially outperforming the standard conditions, which gave a yield of about 60%. In Fig. 2, we summarize the discovery loop.

In this lecture, we use the chemical reaction design as an example to explain Bayesian optimization (BO).

1 From Bayesian Inference to Bayesian Optimization

In a prediction task, our goal is to predict an output y given an input $\mathbf{x}$. Broadly speaking, there are two approaches: **deterministic** prediction and **probabilistic** prediction.

[1] Github: `https://github.com/doyle-lab-ucla/auto-qchem`.

In a deterministic approach, the output is approximated by a single value,

$$\text{Deterministic Prediction:} \quad y \approx f_{\boldsymbol{\theta}}(\mathbf{x}), \tag{1}$$

which provides a point estimate but does not quantify uncertainty.

In contrast, probabilistic prediction treats the output as a random variable and models a conditional distribution,

$$\text{Probabilistic Prediction:} \quad y \sim p(y \mid \mathbf{x}, \mathcal{D}), \tag{2}$$

where $\mathcal{D}$ denotes the *observed data*. Probabilistic models provide both an expected value and an associated **uncertainty**, which is crucial for decision-making in expensive experiments.

Bayesian inference provides a principled framework for probabilistic prediction. The key idea is to represent uncertainty explicitly and *update our beliefs as new data are observed*. In Bayesian regression, we place a probability distribution over the unknown function $f(\mathbf{x})$, meaning that at any input $\mathbf{x}_0$, the output $f(\mathbf{x}_0)$ is a *random variable*.

To start, we specify a **prior** over the unknown function,

$$\text{Prior:} \quad f(\mathbf{x}) \sim p(f), \tag{3}$$

and update this prior using observed data to obtain the **posterior** distribution,

$$\text{Prior: } p(f) \quad \xrightarrow{\text{new data } \mathcal{D}} \quad \text{Posterior: } p(f \mid \mathcal{D}). \tag{4}$$

Predictions at a new input $\mathbf{x}_0$ are then obtained from the posterior predictive distribution,

$$p(y \mid \mathbf{x}_0, \mathcal{D}) = p(f(\mathbf{x}_0) \mid \mathcal{D}), \tag{5}$$

which provides both a mean estimate and an uncertainty.

1.1 Gaussian Process Regression

A Gaussian process (GP) is a Bayesian framework with a **Gaussian process prior**. Given a finite set of feature variables $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, GP assumes that the corresponding function values

$(f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))$ **follow a joint multivariate Gaussian distribution**:

$$p(\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^n \det K}} \exp\left(-\frac{1}{2}(\mathbf{f} - \boldsymbol{\mu})^{\mathrm{T}} K^{-1} (\mathbf{f} - \boldsymbol{\mu}) \right),$$

where $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))^{\mathrm{T}}$ is the vector of function values, $\boldsymbol{\mu}$ is the mean vector, and K is the **covariance matrix**.

The covariance matrix K is defined by a **kernel function**:

$$k(\mathbf{x}, \mathbf{x}'),$$

which encodes correlations between function values at different inputs. The kernel determines how information from one input $\mathbf{x}$ influences predictions at nearby points.

During Gaussian process regression, the kernel function is updated based on newly observed data, which updates the covariance matrix and ultimately refines the posterior distribution over functions.

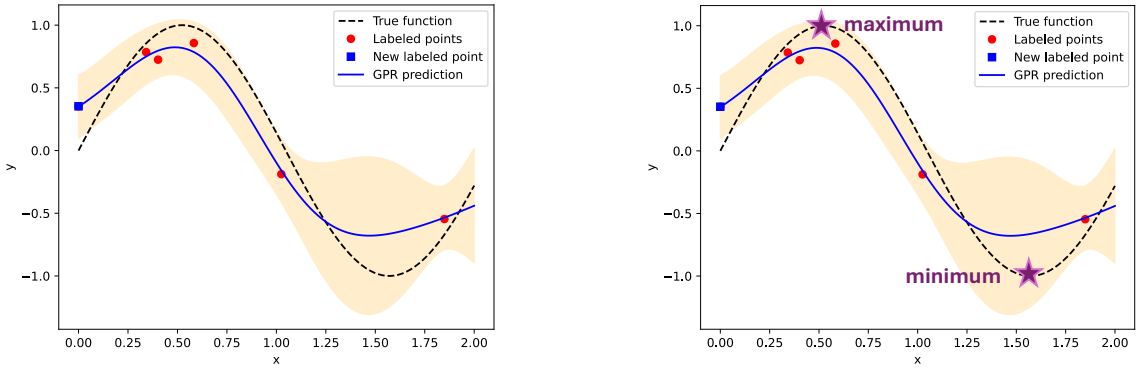
$$\text{New data} \Rightarrow \text{Update hyperparams of } k(\mathbf{x}, \mathbf{x}') \Rightarrow \text{Update } K \Rightarrow \text{Update } p(f) \tag{6}$$

Common choices include the radial basis function (RBF) kernel, which assumes that similar inputs produce similar outputs. Conditioning the GP prior on observed data yields a posterior predictive mean and variance, which serve as the surrogate model for Bayesian optimization. The mean gives a prediction of the objective, while the variance quantifies uncertainty, enabling BO to balance exploration and exploitation when selecting new experiments.

1.2 From Inference to Optimization

As the names indicate, inference means making a prediction, while optimization means **finding the input that maximizes or minimizes a target function**. In Bayesian optimization, instead of predicting outputs y at given inputs, we *seek the input conditions* $\mathbf{x}$ *that maximize (or minimize) a target function*, such as reaction yield, selectivity, or reaction rate, while using as few experiments as possible. An example to compare probabilistic inference and optimization is shown in Fig. 3.

$$\begin{aligned} \mathbf{x}_{\mathrm{BO}} &= \arg\max_{\mathbf{x}} f(\mathbf{x}), \\ \text{where } f(\mathbf{x}) &\sim p(f \mid \mathcal{D}), \end{aligned} \tag{7}$$



(a) Inference. (b) Optimization.

Figure 3: Inference vs. Optimization.

Note that for certain tasks, max can be replaced with min, depending on whether the goal is to maximize or minimize the target.

In the rest of the lecture, we will outline the workflow of Bayesian optimization (BO).

2 Building Blocks

The two main components of BO are the **surrogate model**, which probabilistically approximates the objective function, and the **acquisition function**, which guides the selection of the next input to evaluate.

2.1 Surrogate Model

Given an input $\mathbf{x}$, such as experimental conditions, a surrogate model predicts the target function $f(\mathbf{x})$, for example the reaction yield, along with an associated uncertainty $\sigma(\mathbf{x})$.

$$\begin{aligned} \text{True experiment:} \quad & \mathbf{x} \xrightarrow{\text{Experiment}} y \\ \text{Surrogate prediction:} \quad & \mathbf{x} \xrightarrow{\text{Surrogate}} f(\mathbf{x}) \pm \sigma(\mathbf{x}) \end{aligned} \tag{8}$$

As new experimental data become available, the surrogate model is updated accordingly. The most commonly used surrogate model is the **Gaussian process** (GP), which naturally provides both a predicted mean and uncertainty. Other approaches, such as Bayesian neural networks [5], have also been explored, but they will not be discussed in this course.

2.2 Acquisition Function

Given a surrogate model, the next question is how to select new data points to improve it. This is done using an **acquisition function**, which plays a role similar to the query rule in active learning discussed in the previous lecture.

What criteria should the newly selected data satisfy? Ideally, the next data point (i.e., experimental conditions) should be

- likely to optimize the target function, which depends on the **predicted mean** of $f(\mathbf{x})$;
- informative for improving the surrogate model, which depends on the **uncertainty (variance)** of $f(\mathbf{x})$.

An acquisition function balances the two competing objectives of exploitation (selecting inputs likely to improve the target) and exploration (selecting inputs with high uncertainty to improve the surrogate model). The two most commonly used acquisition functions in chemistry are Expected Improvement (EI) and Upper Confidence Bound (UCB). Another acquisition function, Probability of Improvement (PI), is also used in Bayesian optimization, but it is less effective for typical chemistry tasks and will not be discussed here.

Expected Improvement (EI) is the most widely applied in chemistry, particularly for reaction optimization, catalyst screening, materials discovery, and high-throughput experimentation. Upper Confidence Bound (UCB) is often employed in large or high-dimensional chemical spaces, virtual screening campaigns, and autonomous or closed-loop experimentation platforms.

2.2.1 Expected Improvement (EI)

The idea of expected improvement (EI) is to select the next input $\mathbf{x}$ that is expected to improve upon the current best observed value f_{best}. Define the improvement at $\mathbf{x}$ as

$$I(\mathbf{x}) = \max(f(\mathbf{x}) - f_{\text{best}},\, 0)\,. \tag{9}$$

Since $f(\mathbf{x})$ is not deterministic but follows a predictive distribution $p(f \mid \mu(\mathbf{x}), \sigma^2(\mathbf{x}))$, we define the **expected improvement** as

$$\text{EI}(\mathbf{x}) = \int_{f_{\text{best}}}^{\infty} (f - f_{\text{best}})\, p(f \mid \mu(\mathbf{x}), \sigma^2(\mathbf{x}))\, \mathrm{d}f. \tag{10}$$

When a Gaussian process is used as the surrogate model, the predictive distribution is Gaussian,

$$p(f \mid \mu, \sigma^2) = \mathcal{N}(f \mid \mu, \sigma^2),$$

and the expected improvement admits a closed-form expression,

$$\mathrm{EI}(\mathbf{x}) = (\mu(\mathbf{x}) - f_{\mathrm{best}})\,\Phi(Z) + \sigma(\mathbf{x})\phi(Z), \tag{11}$$

where

$$Z = \frac{\mu(\mathbf{x}) - f_{\mathrm{best}}}{\sigma(\mathbf{x})}. \tag{12}$$

Here, $\phi(z)$ and $\Phi(z)$ denote the probability density function (PDF) and cumulative distribution function (CDF) of the standard normal distribution,

$$\phi(z) = \mathcal{N}(z \mid 0, 1), \quad \Phi(z) = \int_{-\infty}^{z} \phi(t)\, dt. \tag{13}$$

Finally, Bayesian optimization proceeds by selecting the input $\mathbf{x}$ that maximizes the acquisition function,

$$\mathbf{x}_{\mathrm{next}} = \arg\max_{\mathbf{x}} \mathrm{EI}(\mathbf{x}), \tag{14}$$

and evaluating the true experiment at this location.

We see that the definition of EI considers both mean and variance, and therefore satisfies the two criteria of an acquisition function.

2.2.2 Upper Confidence Bound (UCB)

Unlike EI, which is derived from explicit expectations or probabilities, UCB provides a simple deterministic rule that directly balances exploitation and exploration.

The UCB acquisition function is defined as

$$\mathrm{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \kappa\,\sigma(\mathbf{x}), \tag{15}$$

where $\mu(\mathbf{x})$ is the predicted mean of the surrogate model, $\sigma(\mathbf{x})$ is the predictive standard deviation, and $\kappa > 0$ is a user-defined parameter that controls the trade-off between exploitation and exploration.

The two terms in UCB have clear interpretations:

- $\mu(\mathbf{x})$ encourages **exploitation** by favoring regions with high predicted performance.
- $\sigma(\mathbf{x})$ encourages **exploration** by favoring regions where the model is uncertain.

A larger value of κ promotes more exploratory behavior, while a smaller value of κ leads to

more aggressive exploitation. In practice, κ can be fixed, scheduled to decrease over iterations, or tuned based on experimental cost and noise.

In Bayesian optimization, the next experiment is selected by maximizing the UCB acquisition function,

$$\mathbf{x}_{\text{next}} = \arg\max_{\mathbf{x}} \text{UCB}(\mathbf{x}). \tag{16}$$

Compared to Expected Improvement, UCB is simpler to compute and easier to interpret, but its performance depends sensitively on the choice of κ. Nevertheless, UCB is widely used in chemistry and materials science, especially in large or high-dimensional search spaces.

3 Bayesian Optimization Workflow

Now that we have covered the two pillars of Bayesian optimization, we can summarize the overall workflow.

Preparation Step

Before starting the Bayesian optimization loop, we typically determine the following aspects:

- **Input x**: the experimental conditions or design variables to explore.
- **Target function** $f(\mathbf{x})$: the quantity to optimize, e.g., reaction yield, selectivity, or catalytic efficiency.
- **Surrogate model**: the predictive model used to approximate $f(\mathbf{x})$, usually a Gaussian process.
- **Acquisition function**: the criterion for selecting the next experiment, e.g., Expected Improvement (EI) or Upper Confidence Bound (UCB).
- **Stopping criteria**: maximum number of iterations, experimental budget, or target performance.

Warm-up: Initial Dataset

Before entering the iterative BO loop, a small number of initial experiments are performed to provide the surrogate model with some data. These initial points can be chosen randomly or guided by expert intuition.

$$\text{A handful experiments} \rightarrow \text{Initial Dataset} \tag{17}$$

The Bayesian Optimization Loop

1. Fit the surrogate model to the current dataset.
2. Evaluate the acquisition function across candidate inputs.
3. Select the next input $\mathbf{x}_{\text{next}}$ that maximizes the acquisition function.
4. Run the experiment using conditions described by $\mathbf{x}_{\text{next}}$, observe the result, and update the dataset.
5. Repeat steps 1–4 until the target is sufficiently optimized or the stopping criteria are met.

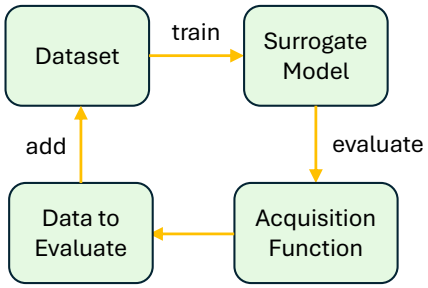


Figure 4: Bayesian optimization loop.

4 Applications

To further illustrate the BO workflow, we list some chemical applications in this section.

4.1 Reaction Optimization (Shields et al., 2021)

- **Input (x)**: choice of solvents, catalysts, ligands, temperature, reagent concentrations, and DFT-encoded features of molecules or catalysts.
- **Target function ($f(\mathbf{x})$)**: reaction yield or selectivity.
- **Surrogate model**: Gaussian process.
- **Acquisition function**: Expected Improvement (EI).
- **Warm-up**: initial reactions randomly selected.
- **BO Loop**: Iteratively propose new reaction conditions predicted to maximize yield; new experimental outcomes are fed back to update the GP.
- **Outcome**: BO rapidly discovers conditions giving 99% yield versus 60% under standard conditions.

4.2 Catalyst Discovery

- **Input (x)**: chemical structure descriptors of candidate catalysts, reaction conditions.
- **Target function ($f(\mathbf{x})$)**: catalytic efficiency or turnover number[2].
- **Surrogate model**: Gaussian process or Bayesian neural network.
- **Acquisition function**: EI or UCB.

[2]Turnover number (TON) is a standard metric in catalysis that measures how many times a single catalyst molecule converts reactant to product before it becomes inactive.

- **Warm-up**: Small set of known catalysts.
- **BO Loop**: Iteratively suggest new catalysts for testing; results feed back into the surrogate.
- **Outcome**: Efficiently identifies high-performing catalysts with fewer experiments than exhaustive screening.

4.3 Materials Design

- **Input (x)**: compositional ratios, synthesis temperature, annealing time, pressure, or dopants.
- **Target function ($f(\mathbf{x})$)**: conductivity, stability, band gap, or other material property.
- **Surrogate model**: GP using composition-property features or latent embeddings from a materials graph.
- **Acquisition function**: UCB for high-dimensional exploration, sometimes combined with EI.
- **Warm-up**: A set of known materials properties from prior experiments.
- **BO Loop**: Suggest new compositions, synthesize them, update surrogate.
- **Outcome**: Identifies optimal material compositions efficiently without exhaustive combinatorial experiments.

4.4 High-Throughput Virtual Screening (HTVS)

- **Input (x)**: molecular structures or SMILES strings encoded as descriptors.
- **Target function ($f(\mathbf{x})$)**: predicted binding affinity or stability.
- **Surrogate model**: Gaussian process or Bayesian neural network predicting property from descriptors.
- **Acquisition function**: EI to prioritize molecules most likely to exceed previous best candidates.
- **Warm-up**: Small subset of known molecules.
- **BO Loop**: Suggest next molecules to evaluate computationally or experimentally; feed results back into model.
- **Outcome**: Dramatically reduces the number of molecules that must be screened to identify top candidates.

4.5 Machine Learning Hyperparameter Optimization

In addition to chemical design, you can also use BO to optimize the hyperparameters of your ML training.

- **Input (x)**: hyperparameters of the model, e.g., learning rate, batch size, number of layers, number of neurons per layer, regularization strength.
- **Target function ($f(\mathbf{x})$)**: validation accuracy, F1 score, or other performance metric.
- **Surrogate model**: Gaussian process or Bayesian neural network predicting performance based on hyperparameters.
- **Acquisition function**: Expected Improvement (EI) or Upper Confidence Bound (UCB) to balance exploration of new hyperparameter regions and exploitation of promising settings.
- **Warm-up**: A small set of randomly sampled hyperparameter configurations.
- **BO Loop**: Iteratively propose new hyperparameter sets to evaluate; feed the observed model performance back into the surrogate to guide the next selection.
- **Outcome**: Efficiently finds near-optimal hyperparameters with far fewer model trainings compared to grid search or random search.

5 Lab

In this Lab session, we will use BO to optimize a chemical reaction, the Buchwald-Hartwig (BH) reaction. The BH reaction is a palladium-catalyzed cross-coupling that forms C–N bonds between an aryl (or heteroaryl) halide and an amine. An example of BH reaction is shown in Fig. 5.

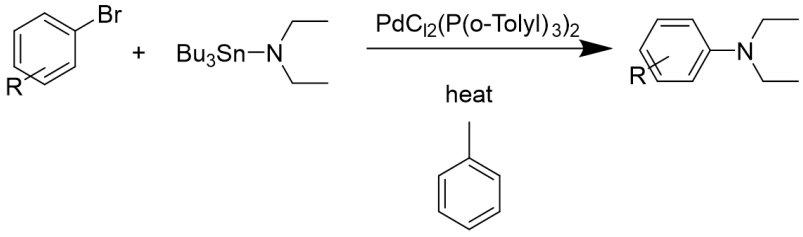


Figure 5: Buchwald-Hartwig (BH) cross-coupling reaction. (Figure derived from LibreTexts.)

We will use a dataset generated by Doyle Group [6], which contains 3,955 BH reactions with various reaction conditions. The columns in the `csv` file is summarized in Table 1. Note that in this dataset, the amine substrate is fixed for each reaction index and is implicitly encoded via the reaction or product label; therefore it is not treated as an independent input variable.

5.1 Implementation

At this point, we can either implement every building block and the BO loop from scratch, or use existing BO packages designed for chemiscal tasks, which I highly recommend if you want to do serious research-level BO tasks.

Table 1: Description of columns in the Buchwald–Hartwig reaction dataset

Column	Description
`reaction`	Reaction identifier: 0,1,2,3,4; Used for bookkeeping or indexing; not used as a model input.
`ligand`	Phosphine ligand coordinated to the Pd catalyst: SMILES string.
`additive`	Optional additive included in the reaction (e.g., salts, acids): SMILES string.
`base`	Base used to deprotonate the amine: SMILES string.
`aryl halide`	Aryl or heteroaryl electrophile, the reactant: SMILES string.
`product`	Coupled aryl amine product: SMILES string. Usually excluded from model inputs to avoid information leakage.
`rxn`	Encoded reaction string: Reaction String (similar to SMILES string). Used for traceability and linking to raw experimental records.
`yield`	Experimental reaction yield: percentage. This is the **target function** $f(\mathbf{x})$ optimized in Bayesian optimization.

However, for the demonstration purpose, we will use built-in functions from a Python Library: `scikit-optimize`: `https://scikit-optimize.github.io/stable/`, installed and called by

- Installation:

```
pip install scikit-optimize
```

- Import:

```
import skopt
from skopt import gp_minimize # the BO module
```

Link to Lab 12:

`https://colab.research.google.com/drive/18719TNs1quHUEEaXcpkuwtthk1O2MXMF?usp=sharing`

References

[1] Stefan Desimpel, Matthieu Dorbec, Kevin M Van Geem, and Christian V Stevens. Bayesian optimization for chemical reactions. *Chemical Society Reviews*, 2026.

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

[3] Yichi Zhang, Daniel W Apley, and Wei Chen. Bayesian optimization for materials design with mixed quantitative and qualitative variables. *Scientific reports*, 10(1):4924, 2020.

[4] Anthony E Klon. Bayesian modeling in virtual high throughput screening. *Combinatorial Chemistry & High Throughput Screening*, 12(5):469–483, 2009.

[5] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29, 2016.

[6] Derek T Ahneman, Jesús G Estrada, Shishi Lin, Spencer D Dreher, and Abigail G Doyle. Predicting reaction performance in c–n cross-coupling using machine learning. *Science*, 360(6385):186–190, 2018.