DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Bayesian optimization of material synthesis parameters

Luisa Ortner

DEPARTMENT OF INFORMATICS

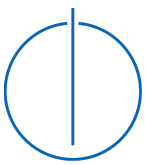
TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Bayesian optimization of material synthesis parameters

Bayes'sche Optimierung von Materialsyntheseparameter

Author:	Luisa Ortner
Supervisor:	Dr. Felix Dietrich
Advisor:	Dr. Manuel Tsotsalas
Submission Date:	15.01.2023



I confirm that this bachelor's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.01.2023 Luisa Ortner

Acknowledgments

First, I would like to thank my supervisor, Dr. Felix Dietrich, for the help and advice and my advisor regarding the chemical part, Dr. Manuel Tsotsalas, Karlsruhe Institute of Technology. I would also like to thank Lena Pilz for performing the synthesis reactions, the always helpful explanations on chemical questions and proofreading the thesis. Thanks also to Josef Mayr for proofreading.

Abstract

Metal-organic frameworks are microporous materials composed of inorganic building units held together by organic molecules. As a result of their construction, there is an enormous variety of compositions, structures, properties and applications. Depending on the substances used in the synthesis, different metal-organic frameworks are obtained. These frameworks have different properties depending on the synthesis parameters, such as temperature, time, and reactants ratio, forming an almost infinite chemical space, and exploring this space requires an extensive amount of sample points. Therefore, it is advantageous to use computational methods instead of experimental methods for the discovery and modification of metal-organic frameworks. However, there is a lack of research on the usefulness of machine learning methods to find parameters for metal-organic framework synthesis, as machine learning is still quite new in the field of chemistry. This bachelor thesis attempts to address this knowledge gap by implementing Bayesian optimization with Gaussian Processes, a machine learning technique, to find the optimal synthesis conditions of metal-organic framework nanoparticles. Bayesian optimization is a global optimization strategy that is particularly useful for black-box functions that are difficult to evaluate. This thesis discusses the process of selecting a suitable kernel for the Gaussian process and applies single-objective and multi-objective Bayesian optimization methods on the example of ZiF-8. The given dataset used to train the model only consists of 30 data points, which presented a challenge for accurate and meaningful predictions. Nevertheless, Bayesian optimization predicted suitable synthesis parameters, as confirmed by experiments at Karlsruhe Institute of Technology. Based on this work, Bayesian optimization for synthesis parameter discovery can be further developed and improved.

Contents

1. Introduction

Much of the progress of today's society and life can be attributed to achievements in material science, as these have enabled new products in a variety of areas, such as energy, health, transportation and the environment. Metal-organic frameworks are one class of materials that have contributed to this progress. There is an enormous number of metal-organic frameworks with a diverse spectrum of properties, all of which are characterized by porosity. Therefore, they find applications in a wide variety of fields, such as energy and gas storage, drug delivery, gas separation and chemical sensors. Metal-organic frameworks can also help to meet future challenges such as the climate crisis, where they have the capability to capture CO_2 [36]. Therefore, it is important to further research and optimize these materials to discover their great chemical potential. Increasing research has focused on the search for synthesis conditions in order to create optimal metal-organic frameworks that meet current needs. Since the discovery of suitable synthesis parameters is difficult, the development of parameter search algorithms is of great importance. Therefore, in this thesis Bayesian optimization and Gaussian processes are used to predict synthesis conditions for metal-organic frameworks to obtain optimal materials.

Metal-organic frameworks consist of metal ions and organic linkers that form three-dimensional structures. By varying their components, previously undiscovered metal-organic frameworks can still be synthesized. Since their structure and properties are determined by their synthesis conditions, it is particularly important to find the optimal set of synthesis parameters. However, this is not trivial because the synthesis parameters form a very large search space which cannot be fully explored experimentally. The use of sophisticated experimental techniques and equipment increases the number of controllable synthesis parameters, expanding the search space of possible synthesis conditions even further. It is difficult, and sometimes impossible, to find the best synthesis conditions in a multidimensional search space using an uncontrolled trial-and-error approach. Therefore, it is necessary to develop more sophisticated approaches to deal with the high dimensionality and complexity of this problem. Machine learning methods have proven to be valuable tools for identifying regions of interest, and several machine learning approaches have been applied for the discovery of optimal sythesis parameters [21]. In this application, it has been shown that early machine learning models already outperform human experts. Machine learning can capture and exploit

even small correlations in the data to improve estimates of synthesis conditions where human experts cannot compete since it exceeds the experts general intuition [21]. Although there has been rapid progress in the field of metal-organic frameworks, the potential of predicting metal-organic framework synthesis conditions using machine learning still has not been fully explored.

Bayesian optimization is a more promising method as it can find the gobal optimum in fewer search steps than other approaches. It is an efficient method to find the global optimum of an expensive to evaluate black-box function. The exploration of metal-organic framework properties requires the optimization of a multidimensional space of synthesis parameters. Bayesian optimization is a method for efficient multidimensional optimization and therefore capable of optimizing the synthesis parameters with a small number of experiments. This is desirable since conducting an experiment is time and resource consuming. Recently it was found useful in the application area of chemistry [15, 13, 4, 14] but has not yet been sufficiently explored [31]. It has been shown that Bayesian optimization outperforms human expert decision making [31]. Some papers differentiate and Bayesian optimization algorithm prevails both in average optimization efficiency, i.e., the number of experiments, and in consistency, i.e., the variance of the result to the initial data set [31].

In chapter 2, the required theory on Bayesian optimization, Gaussian process and metal-organic frameworks is fist explained. In the main part (chapter 3), the metodology of the discovery and optimization of synthesis conditions is discussed in more detail and the background of the metal-organic framework used to perform Bayesian optimization is explained. Afterwards, the two applied approaches single-objective Bayesian optimization and multi-objective Bayesian optimization are explained and evaluated.

2. Related Work

2.1. Gaussian Processes

A Gaussian process (GP) is a data-driven method, which uses a probabilistic approach to machine learning methods. The goal of GPs is to find the function f that best describes the given training data. The advantages of GP include accurate generalization of new and unseen data as well as providing an uncertainty estimation without additional evaluation.

2.1.1. Introduction to Gaussian Processes

Simplified, a stochastic process is a generalization of a probability distribution describing a finite-dimensional random variable to functions. Therefore, a GP is a stochastic process of random variables with a Gaussian distribution. The idea is given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, ..., n\}$, considering all functions which fit to the given dataset. In the following $\mathbf{X}$ will be the input and since $\mathbf{x}$ is a vector, $\mathbf{X}$ can be either a set or a matrix.

Definition 2.1.1 (Gaussian Process)**.** "A Gaussian process is a generalization of the Gaussian probability distribution. Whereas a probability distribution describes random variables which are scalars or vectors (for multivariate distributions), a stochastic process governs the properties of functions [28]."

This definition is the main advantage of GPs, since they combine a sophisticated and consistent view with computational tractability. Generalization of the normal distribution results in the multivariate normal distribution, which samples vectors of numbers instead of numbers. Further generalization leads to a sampling of an infinite vector of numbers, which in other words are functions. This is the approach of a GP, since the properties of the function are only described by finitely many points and the inference is done at these points, it is computationally tractable to compute with functions and it is still consistent with considering all infinitely many points. Therefore, it is not a hindrance that no analytical description of the probability density function of the GP exists in general.

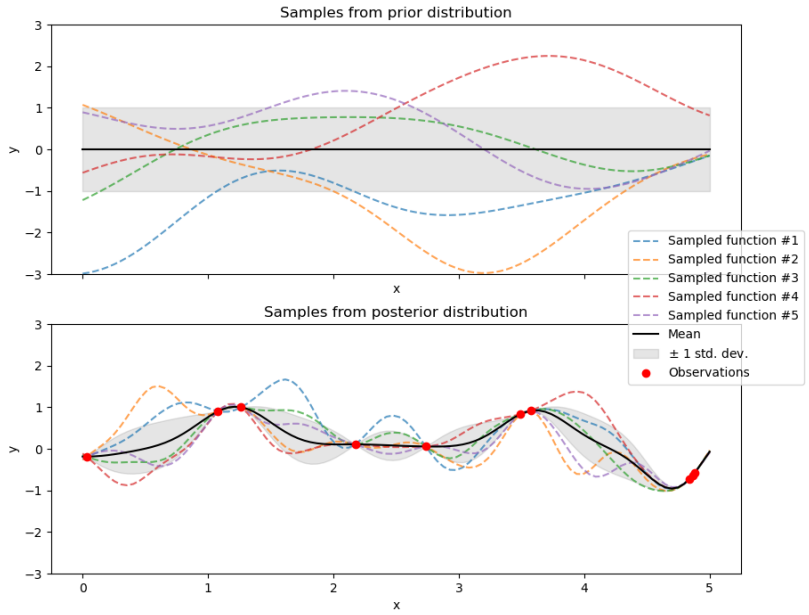


Figure 2.1.: Samples drawn from a GP with RBF kernel adapted from [27].

A GP is a distribution over functions, which can be completely determined by its mean and covariance functions. The GP entirely defines the properties that the functions of its set hold. The above objective, the best function of the set, is defined by the mean of the distribution over functions, with a minimum probability assigned to each function. This can be adjusted by assigning an individually higher probability to some functions that seem more likely based on certain probabilities, e.g. smoother functions [28].

2.1.2. Introduction to Bayesian modeling with Gaussian Processes

In Bayesian inference, the GP begins before conditioning on the data; the GP is a prior distribution over functions and represents the beliefs over the kinds of functions which are expected to be observed. Without any knowledge, the mean over the sample functions at each $\mathbf{x}$ is zero. The posterior distribution over the target functions is obtained using Bayes' theorem on the prior distribution and data. The posterior is then used to make predictions.

In Figure 2.1 samples of a GP are shown for a 1D regression problem. In the first figure the samples are drawn from the prior distribution and in the second figure from

a posterior distribution with eight observations. The standard deviation is independent of x for the prior and is adjusted to be zero at already observed points and high in areas without close function evaluations. The mean is shown in black and represents the prediction of the GP.

2.1.3. Gaussian Process Regression

In this section, a formal definition of the above concept is introduced with regard to the regression problem at hand. Gaussian process regression (GPR) is a supervised machine learning method usually based on a dataset mapped from an input $\mathbf{x}$ to an output $f(\mathbf{x})$, where the function f is continuous. This training data is not absolutely necessary, since the GP can already be evaluated without training.

As stated, the GP is completely specified by its mean function $m(\mathbf{x})$ and covariance function called kernel $k(\mathbf{x}, \mathbf{x}')$

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{2.1}$$

The mean and kernel of a real process $f(\mathbf{x})$ are defined as follows,

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}\left[f(\mathbf{x})\right] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}\left[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))\right]. \end{aligned} \tag{2.2}$$

The mean function is usually set to zero, $m(\mathbf{x}) = 0$, since uncertainty about the mean function can also be accounted for by adding an additional term to the kernel. This is especially done when there is no specific knowledge about a trend in the data. Therefore, the GP is completely determined by its kernel, which specifies how the functions are interpolated between data points. The kernel is a measure of the correlation of two inputs $\mathbf{x}$ and $\mathbf{x}'$. The main difficulty with GPs is to construct a kernel that best represents the particular structure of the data being modeled.

Since the GP is a collection of random variables, it implies consistency, called the marginalization property. This means that the probability distribution of the different sizes of subsets does not change. It is always fulfilled, if the kernel specifies the entries of the kernel matrix

$$\mathbf{\Sigma} = K(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1') & \dots & k(\mathbf{x}_1, \mathbf{x}_n') \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1') & \dots & k(\mathbf{x}_n, \mathbf{x}_n') \end{bmatrix}. \tag{2.3}$$

The kernel matrix represents the covariance between two elements of the training data $\mathbf{X}$. A kernel matrix must be positive definite to be valid, implying that the matrix is symmetric and invertible [28, 1].

GPR learns a generative probabilistic model of the objective function and can thus provide meaningful confidence intervals and posterior samples along with the predictions [27].

2.1.4. Prediction of Gaussian Process Regression

In the following, the theory underlying the prediction of GPR will now be explained. A distinction is made between noise free and noisy observations.

Prediction using Noise free Observations

Given a training dataset with noise free function values $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ and inputs $\mathbf{X}$, the prior has to be transformed in the posterior to make predictions $\mathbf{f}_*$ at new inputs $\mathbf{X}_*$. The joint distribution of the observations $\mathbf{f}$ and the predictions $\mathbf{f}_*$ is

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right). \tag{2.4}$$

To obtain the posterior distribution over the functions, the joint Gaussian prior distribution needs to be constrained to contain only functions which fit the observed data points. This can be done by conditioning the joint Gaussian prior distribution on the observations:

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{X}, \mathbf{f}, \mathbf{X}_*) &= \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*), \quad \text{where} \\ \boldsymbol{\mu}_* &= K(\mathbf{X}_*, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}\mathbf{f}, \\ \boldsymbol{\Sigma}_* &= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})K(\mathbf{X}, \mathbf{X})^{-1}K(\mathbf{X}, \mathbf{X}_*). \end{aligned} \tag{2.5}$$

The function values $\mathbf{f}_*$ can now be sampled from the joint posterior distribution by evaluating the mean and covariance matrix from equation 2.5. It should be noted, that the covariance only depends on the training data and not on the observered test points.

Prediction using Noisy Observations

In an experimental setting, it is assumed that there is normally distributed additive noise in the output data $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, which distinguishes the observational model from the real model:

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon. \tag{2.6}$$

Therefore, the prior is now given by $K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I$ and the predictive distribution is given by

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*) &= \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*), \quad \text{where} \\ \boldsymbol{\mu}_* &\triangleq \mathbb{E}[\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*] = K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{y}, \\ \boldsymbol{\Sigma}_* &= K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I]^{-1}K(\mathbf{X}, \mathbf{X}_*). \end{aligned} \tag{2.7}$$

In practical implementations, most often the posterior means and variances are calculated using Cholesky decomposition and then the linear system of equations is calculated, rather than using Equation 2.5 and inverting the matrix directly, as this is faster and numerically more stable [28].

2.1.5. Kernel functions

The kernel function $k(\mathbf{x}, \mathbf{x}')$ represents the characteristics of the process, such as smoothness, characteristic length-scale, and stationarity. The space of kernels can be divided into two different classes: stationary and non-stationary kernels. Stationary kernels depend only on the distance between two data points and not on their absolute value $k(\mathbf{x}_i, \mathbf{x}_j) = k(d(\mathbf{x}_i, \mathbf{x}_j))$ and are therefore invariant to translations in the input space. Non-stationary kernels additionally depend on the specific values of the data points.

The kernel encodes assumptions about the function to be learned. The value of the kernel function $k(\mathbf{x}, \mathbf{x}')$ is an indicator of the interaction between two states $\mathbf{x}$ and $\mathbf{x}'$ [8].

Kernels determine the shape of prior and posterior of the GP, and therefore, the covariance of the GP between data points. Pre-existing knowledge about the kernel function, e.g. the shape of the objective function, can be exploited by using specialized kernel functions [12, 27].

The Radial Basis Function (RBF)

The radial basis function (RBF), also known as the squared exponential (SE) function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{2l^2}\right) \tag{2.8}$$

only depends on the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. The kernel expresses that the correlation of points decreases with the square of the distance between these points. The parameter $l > 0$ defines the length scale of the kernel, with smaller or larger length scales leading to smaller or larger variations in the resulting functions, as can be seen in Figure 2.2. $d(\cdot, \cdot)$ is the Euclidean distance for the n-dimensional space. This kernel is stationary and infinitely differentiable, meaning that GPs with this kernel have mean square derivatives of all orders and are thus very smooth. The RBF kernel is commonly used as the default kernel [27, 7].

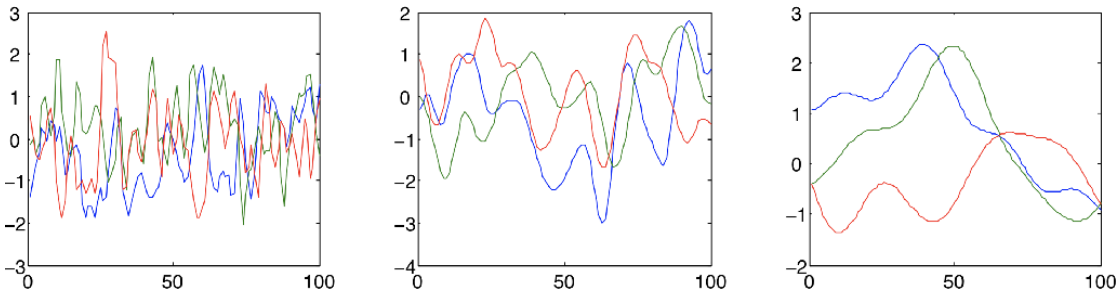


Figure 2.2.: Random functions drawn from a GP prior with the RBF kernel. Each plot has a different length scale that decreases from the left plot to the right plot. Taken from [8].

The Matern Kernel

The Matern kernel [27] is a generalization of the RBF kernel and its equation is given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{l} d(\mathbf{x}_i, \mathbf{x}_j) \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{l} d(\mathbf{x}_i, \mathbf{x}_j) \right). \tag{2.9}$$

$K_\nu(\cdot)$ represents a modified Bessel function and $\Gamma(\cdot)$ the gamma function $\Gamma(m) = (m-1)!$.

Compared to the RBF kernel, the Matern kernel has an additional parameter $\nu = n + \frac{1}{2}, n \in \mathbb{N}$, which controls the smoothness of the resulting functions. With $\nu \to \infty$ the kernel is equivalent to the RBF kernel. A small ν leads to less smooth functions. The kernel represents n-times differentiable functions. Important ν-values are $\nu = 1.5$ (once differentiable) and $\nu = 2.5$ (twice differentiable).

For $\nu = 0.5$ the kernel becomes identical to the absolute exponential kernel, which has the following definition:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{l} \right). \tag{2.10}$$

In contrast to the squared exponential kernel, which is smooth, the absolute exponential kernel is only continuous, thus it is not differentiable. The function approximations generated by kernel methods inherit the smoothness of the kernel, which has implications for modeling. A smooth kernel is suitable for fitting smooth functions, while a differentiable kernel is suitable for differentiable functions [3].

The Rational Quadratic Kernel

The Rational Quadratic kernel [27] can be considered as a scale mixture of RBF kernels with different characteristic length scales. A scale mixture is an infinite sum. Therefore, functions drawn by this kernel vary smoothly across many length scales. It is characterized by the length scale parameter $l > 0$ and the scale mixture parameter $\alpha > 0$, which determines the relative weighting of large and small scale variations. Its definition is given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \frac{d(\mathbf{x}_i, \mathbf{x}_j)^2}{2\alpha l^2}\right)^{-\alpha}. \tag{2.11}$$

ExpSineSquared Kernel

The ExpSineSquared kernel [27] models periodic functions. It is parameterized by the length scale $l > 0$ and a periodicity parameter $p > 0$ and is given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{2\sin(\pi d(\mathbf{x}_i, \mathbf{x}_j)/p)^2}{l^2}\right). \tag{2.12}$$

Other Kernels

There are also a number of other kernels [1, 27, 28], such as the White kernel, which controls the noise component of the signal and can be used as part of a sum kernel, the Constant Kernel, which is mostly used in a sum kernel to modify the mean of the Gaussian process, the Dot-Product kernel, which can be obtained from Bayesian linear regression. A special Dot-Product kernel is the linear kernel.

Kernel Operations

Several kernels can be combined by kernel operations to one kernel [1, 27, 28]. Kernel operations which combine two kernels K_1, K_2 include the Sum kernel, which is defined as

$$K_{\text{sum}}(\mathbf{X}, \mathbf{Y}) = K_1(\mathbf{X}, \mathbf{Y}) + K_2(\mathbf{X}, \mathbf{Y}), \tag{2.13}$$

and the Product kernel being defined as

$$K_{\text{product}}(\mathbf{X}, \mathbf{Y}) = K_1(\mathbf{X}, \mathbf{Y}) \cdot K_2(\mathbf{X}, \mathbf{Y}). \tag{2.14}$$

The exponentiation kernel takes one base kernel and a scalar parameter p and connects them via

$$K_{\text{exp}}(\mathbf{X}, \mathbf{Y}) = K(\mathbf{X}, \mathbf{Y})^p. \tag{2.15}$$

2.1.6. The Log Marginal Likelihood

The marginal likelihood is generally used to provide a measure of model fit and therefore enable comparison between models. It can be derived from the marginal likelihood, which is the marginal over the set of parameters that determine the model and is defined as the integral of the likelihood times the prior:

$$p(\mathbf{y}|\mathbf{X}) = \int p(\mathbf{y}|\mathbf{f},\mathbf{X})p(\mathbf{f}\,|\mathbf{X})\,d\mathbf{f}. \tag{2.16}$$

It is in general not calculateable in closed form, but an approximation can be found using the sum of the full probability and a penalty term. The logarithmic marginal likelihood is specified by

$$\log p(\mathbf{y}|\mathbf{X}) = \underbrace{-\frac{1}{2}\mathbf{y}^\top(K+\sigma_n^2 I)^{-1}\mathbf{y}}_{1}\overbrace{-\frac{1}{2}\log|K+\sigma_n^2 I|}^{2}\underbrace{-\frac{n}{2}\log 2\pi}_{3} \tag{2.17}$$

where σ_n^2 is the noise variance and $K = k(\mathbf{X},\mathbf{X})$. The term consists of a sum of three parts. (1) is the only one that depends on the output data and thus represents the data fitting. (2) penalizes the complexity depending on the kernel and the input data and (3) is the normalization constant. It can be observed that $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, K+\sigma_n^2 I)$. Small or negative values indicate a good model fit [28].

2.2. Bayesian Optimization

Bayesian optimization (BO) has proven useful for globally optimizing black-box objective functions that are difficult to evaluate, with the advantage of tolerating noise in the function evaluations. Thus, Bayesian optimization results in a point that minimizes or maximizes the function f. BO is part of a class of methods called surrogate models, which include all approaches for optimizing objective functions where it is not possible to efficiently evaluate a function point. It has proven to be very powerful and has excellent performance. Several paper [26, 21, 31] showed that BO outperforms both experts and other modern global optimization algorithms.

BO can be applied to various search spaces that include arbitrary parameterized response domains and continuous domains, optimally with less than 20 dimensions. The objective function f is typically difficult, time consuming, or even impossible to evaluate directly. Therefore, BO uses adaptive sampling to reduce the required function evaluations. Moreover, BO is usually used when there is no known structure in f, such as concavity or linearity, otherwise other optimization methods might be

more efficient regarding the number of function evaluations. BO consists of two main components: GPR as the method for statistical inference, and an acquisition function that decides where to draw the next sample. A surrogate is created for the objective function and the GPR is used to quantify the uncertainty in this surrogate. The GP has the advantage that it has a quantification of predictive uncertainty which is used by the acquisition function to get the next best sampling position. Other methods for creating this prediction model for the objective function than GPR include random forest [34] or deep neural networks [32], but will not be considered in this work.

BO is used in many areas, such as tuning hyperparameters in machine learning algorithms, especially deep neural networks, reinforcement learning, calibration of environmental models, cognitive science and chemistry.

2.2.1. Introduction to Bayesian Optimization

Since the objective function is difficult to evaluate, the goal of BO is to approximate the global optimum well enough while minimizing the number of evaluations of the objective function. Therefore, a probabilistic surrogate model for modeling the objective function is created. This Bayesian statistical model is a GP and is compared to the objective function easy to evaluate. For BO an initial dataset is beneficial, which is used to train the GP. This initial dataset can be obtained by evaluating the objective according to an initial space-filling experimental design, often consisting of uniformly randomly selected points. Like in section 2.1 explained, the GP is constructed from prior observations and a prior over functions that capture our assumptions about the response surface. Assumptions could be about the smoothness or experimental noise, but are not always available, which is why several kernels often have to be evaluated. The fitted GP can be evaluated at a candidate point $\mathbf{x}$ to obtain possible values for the objective function $f(\mathbf{x})$.

After training the surrogate model an acquisition function is used to select the next position for evaluating the objective function. The acquisition function uses the GP to determine the expected utility of candidate experiments. After evaluation the objective function at the suggested point, which is obtained by optimizing the acquisition function, the dataset and the GP posterior gets updated. Repeating the process leads to improved results [8, 31, 12].

In the following the BO algorithm is shorty summarized:

1: Train GP with initialization data.
2: **loop**
3: Update posterior prabability distribution of GP with all available data.
4: Optimize acquisition function based on the current posterior distribution.
5: Observe the objective function at the optimized point.
6: **end loop**
7: **return** Position of best value.

Four iterations of BO using GPR for one-dimensional inputs can be seen in Figure 2.3. Each row represents one iteration, and the left panels show the objective function in red with red circles representing the observations. The red shading represents the noise in the objective function. Also shown in green is the output of the GPR. The green solid line is the mean $\mu_{GP}(x)$ of the underlying GP and the green shadow represents the variance $\sigma_{GP}(x)$ of the GP at some point x. The mean can be interpreted as the point estimate of the objective function at x and interpolates the previously evaluated points. In unexplored areas, the uncertainty of the GP is high and thus zero in the observation points because the variance predicts the uncertainty of the model. Both the mean and variance are updated with each subsequent evaluation. On the right side, the corresponding acquisition functions for each iteration after the GP is fit, are shown. The acquisition function represents the most promising areas based on the given knowledge. Here, the expected improvement is used, which will be explained below. The blue circle indicates the next evaluation point and is always the maximum of the acquisition function. At points that have already been observed, the acquisition function is low, since these points are already known and should not be selected again.

2.2.2. Acquisition Functions

The acquisition function $u(\mathbf{x})$ can efficiently determine the next sampling position since the GP is used instead of the objective function, which is inexpensive to evaluate. When selecting the next sample, two relevant criteria are balanced against each other: exploration and exploitation. Exploration aims to explore areas where the uncertainty about the objective function $\sigma(\cdot)$ is high and thus in unexplored areas, whereas exploitation uses the previously gained knowledge and selects a point that is assumed to be closer to the global optimum and thus $\mu(\cdot)$ is high. To achieve the goal of obtaining the best possible value with as few function evaluations as possible, the optimal balance of exploration and exploitation is required [12].

There is a variety of different acquisition function, each has a slightly different goal when selecting the next sample point. There are acquisition functions that are optimal in their minimum, but also those that are optimal in their maximum. They can be

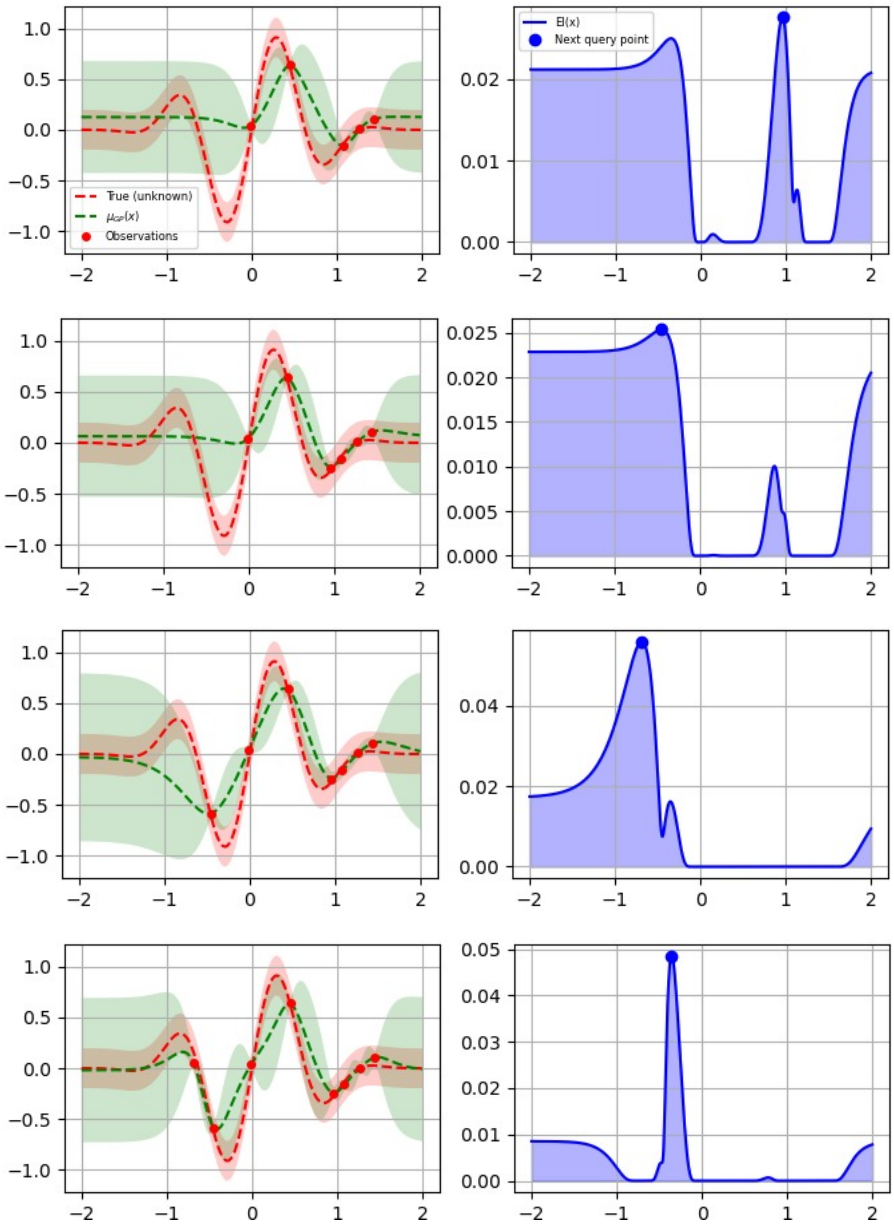


Figure 2.3.: Four Bayesian optimization steps. On the left the true objective function (red) with observations (red dots), mean (green) and covariance function (light green) and on the right the acquisition function. Taken from [16].

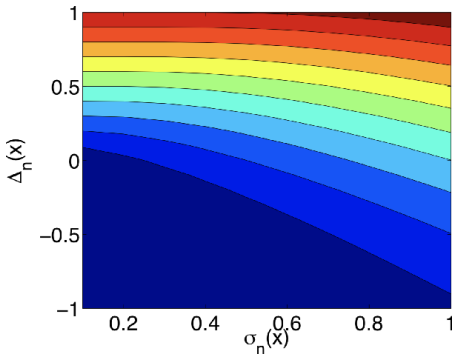


Figure 2.4.: EI contour plot, in terms of $\Delta_n(\mathbf{x})$, the expected difference between the proposed point and the best point $f(\mathbf{x}')$, and the posterior standard deviation $\sigma_n(\mathbf{x})$. Red indicates high and blue smaller values. Taken from [8].

transformed into each other by negation. Most acquisition functions have a parameter κ, which controls the exploration-exploitation trade-off.

Probability of Improvement (PI)

The negative Probability of Improvement (PI) prefers the strategy exploitation by prefering areas with a high probability for improving the current maximum $f(\mathbf{x}') = \max_{m \leq n} f(\mathbf{x}_m)$, where n is the number of evaluations [16]. It is given by

$$-PI(\mathbf{x}) = -P(f(\mathbf{x}) \geq f(\mathbf{x}') + \kappa). \tag{2.18}$$

Expected Improvement (EI)

The expected improvement (EI) is the most common acquisition function and is calculated using the previously evaluated point with the best observed value $f(\mathbf{x}')$. By assuming that only one additional evaluation is possible, EI is derived. The evaluation at the position $\mathbf{x}$ leads to the observation $f(\mathbf{x})$. The value of the best point is then either $f(\mathbf{x})$ or $f(\mathbf{x}')$, and the improvement is then $[f(\mathbf{x}) - f(\mathbf{x}')]^+$, where $a^+ = \max(a, 0)$. Therefore, $\mathbf{x}$ should be chosen to maximize the improvement, and thus the negative expected improvement is calculated by

$$-EI(\mathbf{x}) = -\mathbb{E}[f(\mathbf{x}) - f(\mathbf{x}')]. \tag{2.19}$$

The expected improvement is usually calculated using integration by parts, described in more detail in [8].

In Figure 2.4, the contours of EI are plotted in terms of $\Delta_n(\mathbf{x})$ and the posterior standard deviation $\sigma_n(\mathbf{x})$ of the n-th iteration at the proposed point. $\Delta_n(\mathbf{x}) = \mu_n(\mathbf{x}) -$

$f(\mathbf{x}')$ is the expected quality difference between the proposed point $\mathbf{x}$ and the best point so far. Blue represents small values, while red indicates high values. As a result, the graph shows that $EI_n(\mathbf{x})$ increases in both $\Delta_n(\mathbf{x})$ and $\sigma_n(\mathbf{x})$, and lines with constant EI show, how EI balances between exploitation (high expected quality and therefore high $\Delta_n(\mathbf{x})$) and exploration (high uncertainty and therefore high $\sigma_n(\mathbf{x})$). Evaluation at points with high $\Delta_n(\mathbf{x})$ is desirable because these points are likely to be good approximations of global optima. In contrast, points with high uncertainty provide more information about the target in areas that are not well explored. Therefore, the tradeoff between exploration and exploitation is considered by the EI.

In Figure 2.3, EI is used as acquisition function and is shown in the right panels. Exploitation is used in the first and last figures, while exploration is preferred in the second and third figures [16, 8].

Lower Confidence Bound (LCB)

By using the Lower Confidence Bound (LCB), the trade-off between exploitation and exploration can be controlled by the parameter κ [16].

$$LCB(\mathbf{x}) = \mu_{GP}(\mathbf{x}) - \kappa\sigma_{GP}(\mathbf{x}) \tag{2.20}$$

2.2.3. Bayesian Optimization for Material Synthesis

Bayesian optimization has been found useful in the fields of chemistry, chemical engineering, materials design, and drug discovery, where it takes years and many resources to fully implement BO because it requires repeated experimentation. Some research has already been done, but the number of researchers in these fields who are aware of the benefits of BO is still quite small [8].

BO can be used to predict synthesis parameters for a certain material. The number of evaluations required to build a model is related to the number of parameters. That is, with a large number of parameters, the number of observations also increases, making the creation of the model more complex and challenging. The number of parameters for material science is mostly managable, but depends on the material under consideration. Repeating the process of BO leads to improved results and the process stops when the reaction yield is maximized, the resources are exhausted, or the reaction space is sufficiently explored.

In classical experimental design, modeling and optimization are two different processes. However, in BO, a potentially more efficient sampling is performed by adapting to previous evaluations. Thus, optimization is integrated into the modeling process. Since the goal in the discovery of new materials is to minimize the required resources and thus the number of experiments, BO is suitable because its advantage is a minimal

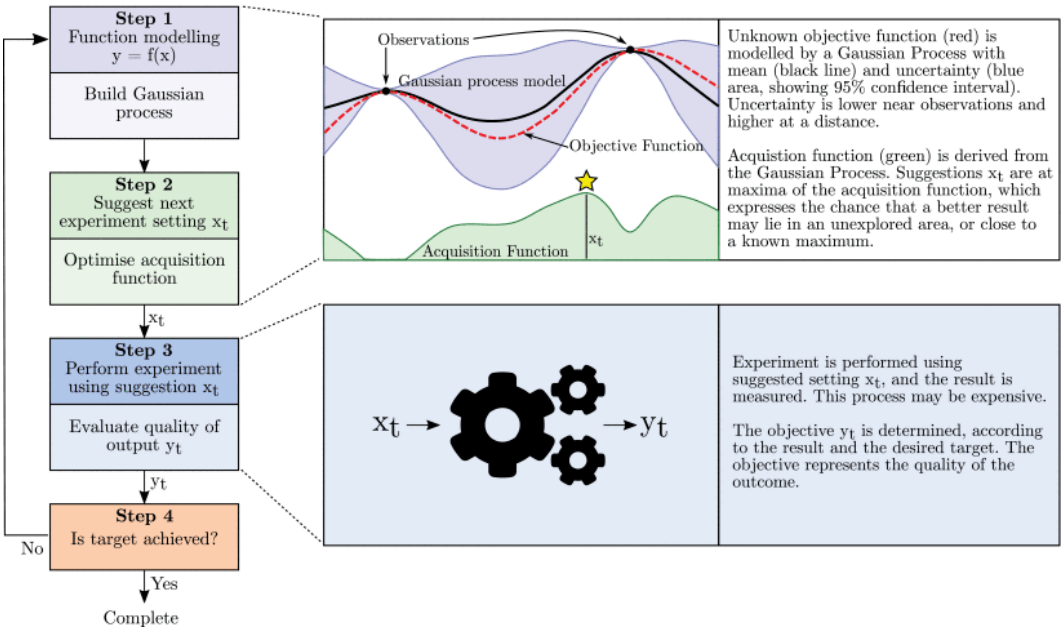


Figure 2.5.: Overview of BO cycle. It is an iterative process in which the unknown black-box function is modeled as a GP, based on which the acquisition function selects the most promising parameter. As more values of the objective function become available over time due to measurements of the performed experiments, the model quality improves. Taken from [12].

number of functional evaluations by using a model-based approach with an adaptive sampling strategy.

Figure 2.5 shows the cycle of BO for experimental applications. Since there exists no objective function, the experiment must be performed after each proposal of the acquisition function. The GP models the belief about the experiment based on the given knowledge, and based on this, the acquisition function selects the next sample point and is thus cheap to compute, unlike the experiment. These synthesis parameters are then used to run and evaluate an experiment, which is then used to further train the GP in the next step [12].

2.2.4. Multi-objective Bayesian optimization

So far, optimization problems had a single objective according to which the optimization was performed. However, in most optimization problems, optimization is not performed according to one objective, but according to several objectives simultaneously.

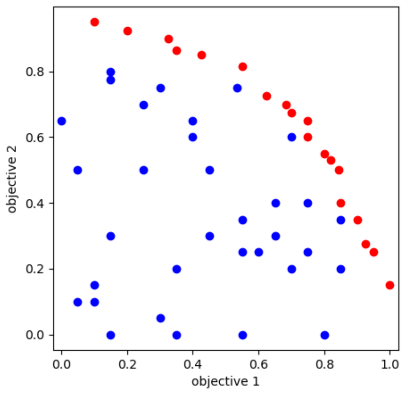


Figure 2.6.: Pareto front in red for two objectives.

This problem is approached by multi-objective Bayesian optimization (MOBO), which works similarly to single-objective Bayesian optimization (SOBO).

In MOBO, the n-dimensional input space $\mathbf{X}$ remains the same as in SOBO, but contrary, the m-dimensional target space $\mathbf{Y}$ is not bounded by $m = 1$. m is defined by the number of targets. $\mathbf{x} = (x_1, x_2, ..., x_n) \in X$ is called the decision vector and $\mathbf{y} = (y_1, y_2, ..., y_m) \in Y$ the target vector [33]. Therefore, the multi-objective optimization problem is defined as

$$\max_{\mathbf{x} \in X} \text{ (or } \min_{\mathbf{x} \in X}) \ \mathbf{y} = (y_1, y_2, ..., y_m) = f(\mathbf{x}). \tag{2.21}$$

Pareto Front

The problem with the optimization of multiple objectives is that they are usually in conflict with each other, i.e. optimal regions of the different objectives are located in different regions of the search space. Thus, optimizing one objective function leads to the other objectives not being optimal. Therefore, the solution to the optimization problem is a set, called the Pareto front, which is the tradeoff between the objectives. A Pareto front for a problem with two objectives can be seen in Figure 2.6. None of the objectives of the Pareto front can be further improved without degrading the other objective [10].

Definition 2.2.1 (Pareto-dominance)**.** For any two decision vectors $\mathbf{a}$ and $\mathbf{b}$,

$$\mathbf{a} \succ \mathbf{b} \ (\mathbf{a} \text{ dominates } \mathbf{b}) \text{ if } \mathbf{f}(\mathbf{a}) \geq \mathbf{f}(\mathbf{a}), \tag{2.22}$$

$$\text{where } \mathbf{f}_i(\mathbf{a}) > \mathbf{f}_i(\mathbf{b}) \text{ for at least one component } i \text{ of } \mathbf{f}. \tag{2.23}$$

The comparisons between vectors are made elementwise.

The Pareto set is given by

$$\{\mathbf{x}^\star\} = \{\mathbf{x}^* \in X \mid \mathbf{x}^* \succ \mathbf{x}, \forall \mathbf{x} \in X\}. \tag{2.24}$$

Therefore, it contains exactly all non-dominated points of the objective space. The Pareto front is the image of the Pareto set:

$$\mathcal{F} = \{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in \{\mathbf{x}^\star\}\}. \tag{2.25}$$

There exist different approaches to approximate the Pareto set, most of them need a large number of objective function evaluations, which is not ideal in the case of predicting synthesis parameters. MOBO deals with highly expensive objective functions and minimizes the number of function evaluations, which is the simulation bottleneck of the problem, as explained above. Therefore, an approximation of the Pareto set is obtained by the BO.

MOBO uses a separate GP for each objective. In every iteration, after MOBO proposes the synthesis parameters and updates the GPs, the GPs are used to compute an approximation of the Pareto set. Since now no single value is predicted, but several, BO cannot determine an optimal value, but only decide which decision vectors are included in the Pareto set. In the end, the approximated Pareto set is returned instead of the optimal point. There are several approaches for computing such approximations. In this work, the non-dominated sorting genetic algorithm II (NSGAII) [5] is used [10].

Acquisition function

As with SOBO, there are several different acquisition functions in MOBO. In this work, only the acquisition function described in [10] is discussed. It concentrates on improving the Pareto front and represents a compromise between exploitation and exploration.

Mainly points from the approximated Pareto set are selected. The method selects the point on the approximated Pareto set that is farthest from all previously observed points. Then, a randomly selected element of the decision vector with probability $r \in [0, 1]$ is replaced by a value from the respective allowed interval to ensure exploration.

2.3. Metal-organic frameworks

From 1990 onwards, there has been an increased amount of research on materials with porous structures, based on metal ions and organic bridging linkers, which are called

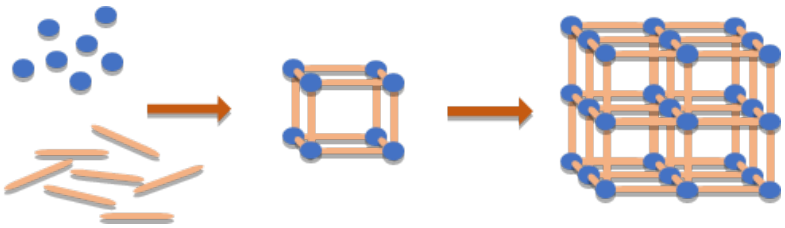


Figure 2.7.: Synthesis of a MOF kindly provided by Lena Pilz.

metal-organic frameworks, or MOFs for short. Some papers also refer to MOFs as coordinating polymers, although some authors use the term MOFs only for structures which exhibit porosity. A large surface area and tunable pore size, which can result in porosity of up to 90%, are generally characteristic of these crystalline materials.

As shown in Figure 2.7, a MOF is a porous 3-dimensional framework of metal nodes (blue spheres) and organic bridging linkers (yellow struts), that create pores inside. This property of MOFs, makes them interesting in the first place, because one can store, transport, and separate other substances. The linker is responsible for the size of these pores, whereby a longer linker creates a larger pore, and thus larger molecules can be stored in it. To support the formation of the MOF, modulators and additives are usually added. The modulator "modulates" the reaction, there are several possibilities how this can happen, e.g. by blocking binding sites. In general, modulators can either facilitate the nucleation process or interfere with particle growth.

There is a wide range of inorganic metal components and an almost infinite choice and design of organic linkers, leading to a variety of material structures and properties. The sum of the physical properties of the inorganic and organic components and the possible synergistic interactions between the two lead to the large number of MOFs. In addition, depending on the respective parameters of the synthesis, such as temperature, time, metal, modulator ratio and reactant ratio, different properties of a MOF are formed during the reaction. As a result, a wide range of structural, magnetic, electrical, optical, and catalytic properties can be introduced into such materials. Recently, there has been a lot of interest in discovering new MOFs. The advantageous properties of MOFs include extremely high surface area, adjustable pore size, and flexible functionality. Creating porosity in these polymeric metal-organic structures has been a challenging but crucial aim in the advancement of this field, as it offers possiblities for chemical separations, ion exchange, sensing, and possibly even catalytic behavior [18, 37].

Their applications range widely, including catalysis and chemical sensors, removal of absorption and separation of toxic substances from gas and liquid, storage of clean energies and environmental applications, like energy and gas storage, and medical and biological applications, like novel drug delivery systems [29, 19, 37].

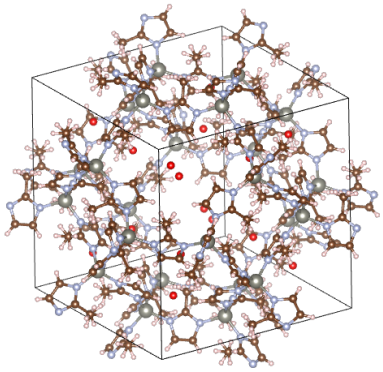


Figure 2.8.: Three dimensional structure of ZiF-8, created with [23].

There is also a varity of challanges in the fabrication and application of MOFs, including brittleness, cracking and the scalability of the process. Besides, the reproducability of identical properties is challenging, since it is difficult to obtain completely identical synthesis conditions. This is due to the circumstance that some of them vary naturally, e.g. the temperature in the room where the synthesis takes place [18, 37].

Although over 100,000 MOFs have already been discovered [9], including only predicted but not yet synthesized MOFs, there is still an incredible number of undiscovered MOFs, where the number cannot be determined. The goal of this work is therefore to optimize the process of the parameter discovery.

In this thesis, the focus will be on three MOF structures to demonstrate Bayesian optimization of material synthesis: ZiF-8, UiO-66 and NH_2-UiO-66. Since MOFs are very diverse, it is generally not possible to draw conclusions regarding good synthesis parameters from one subclass to another.

ZiF-8 belongs to the subclass of zeolitic imidazolate frameworks (ZiFs). ZiFs represent a unique and well researched class since the network topology and associated properties vary widely while maintaining core chemical connectivity. They combine properties and thus advantages of both MOFs and zeolites, which are often compared to MOFs. These include crystallinity, porosity, and superior chemical and thermal stability. ZiF-8 is one of the best developed MOF structures, because of its easy, controllable, and cheap sythesis leading to structural flexibility, high chemical and thermal stability, good gas separation performance, and large specific surface area. Since they can also be manufactured to be biocompatible, they are likewise used for bone fabrication [17]. ZiF-8 has a microporous structure composed of zinc ions ($Zn(ac)_2 \cdot 2\,H_2O$) coordinated with 2-methyl imidazolate linkers forming a solidate zeolite topology. This structure features a space group of interconnected six-membered annular windows, as shown in Figure 2.8. In this work sodium hydroxide (NaOH) is used as modulator and methanol

(CH_3OH) as solvent. Due to its properties such as large surface area and robustness, ZiF-8 finds application in gas separation processes by absorption, membranes, sensing and catalysis, among others [2, 20].

UiO-66 consists of zirconium metal nodes obtained from zirconium chloride (ZrCl4) as well as benzene-1,4-dicarboxylic acid (BDC) as linker. In this work acetic acid (CH_3COOH) and water (H_2O) are used as modulator and the solvent is propanamide (C_3H_7NO). UiO-66 has gained scientific popularity due to its large surface area and high thermal stability, which is attributed to the fact that the metal oxide is cuboctahedral. In addition, it has exceptional tunability and functionality and can therefore be used for many different applications. It can be easily synthesized on a laboratory scale and has high mechanical, acid, aqueous, and water vapor stability. Applications of UiO-66 include catalysis, photocatalysis, adsorption separation, sensors, and biomedicine [35].

NH_2-UiO-66 has in principle the same structure as UiO-66 and differs only in one functional group, at the linker, the amino group. The metal, which is zirconium, the modulator, and the solvent are the same in both cases. This makes the systems very similar, both in structure, properties and applications, because the pore size is not really affected by this modification. However, the functionality of the amino group is added; for instance, something can be specifically bound to it.

3. Bayesian optimization of material synthesis

3.1. Introduction to predicting synthesis conditions

The overall goal of this thesis is to implement BO capable of predicting new good synthesis parameters, while minimizing the number of time- and resource-intensive experiments required. In this way, the time spent on unsuccessful experiments should be significantly reduced. BO predicts the relationship between the synthesis conditions and the fitness metric of the resulting MOFs based on the available data and the sequential search for the global optimum within the search space. The fitness metric is a criterion for the quality of the reaction and is determined by experiments. Since the selection of the kernel and its hyperparameters is crucial for BO, this process has received special attention and has taken a considerable amount of time.

Due to the large search space and the lack of a priori knowledge about the location of good synthesis parameters, predicting the synthesis conditions is a difficult search problem. In MOF synthesis, the linker and metal nodes are often simple enough that the synthesis itself is not the bottleneck. It should be noted that many syntheses are not feasible, and the number of syntheses should therefore be kept as small as possible. Instead, the real challenge is finding the optimal synthesis conditions which result in MOFs with the best properties, like large surface area, stability, and porosity. In this work, the desired properties are uniform nanoparticle size and crystallinity. The understanding of the synthesis process of MOFs, such as crystal growth, kinetics, and energetics of framework bond formation, etc., is too limited to make reliable predictions about good synthesis parameters. A synthesis experiment must first be conducted, which takes a considerable amount of time, to reliably determine the properties of the respective MOF, since the exact geometry is typically unkown a priori. Since MOFs involve many different and numerous chemical compounds, even the known synthesis conditions for a MOF are generally not transferable to new MOFs. As a result, no general synthetic pathway exists for MOFs [25].

The parameters of a typical MOF synthesis include the choice of solvents and their composition, temperature, reaction time, amount of metal, reactants ratio, and modulator ratio. Since there are numerous parameters, the search space grows exponentially

with the number of categorical components, therefore some parameters are held constant. In this work the choice of solvent, metal source, modulator, solvent, and solvent volume is considered as fixed, while the temperature, time, metal, reactants ratio, and modulator ratio are variable parameters whose value must be optimized. More about these variable parameters can be found in section 3.3 Dataset. This results in a high-dimensional chemical space that must be searched for the parameters that lead to the desired formation, crystallisation and properties of the MOF.

3.2. Approaches for predicting synthesis conditions

In the following, the different approaches for predicting synthesis conditions are presented.

3.2.1. Classical approach

The classical discovery of reaction parameters for experiments is usually done as follows. Chemists search the chemical literature for similar reactions and deduce the most influential dimensions for reaction success based on experience, mechanistic understanding, empirical data and simple heuristics. Then they perform experiments, changing only one parameter at a time, because the parameters might influence each other. Afterwards, the resulting MOF is analyzed and based on that the next synthesis parameters are selected. This leads to a large number of experiments, which is very time and resource consuming. However, since only a small subset of all possible synthesis parameters can be performed in practice, finding the best experimental conditions is not guaranteed, and chemists must hope to obtain reasonably good results [31].

Several algorithms for finding MOF synthesis parameters have already been explored, among them grid search, which is an approach to search the chemical space without any prior knowledge. However, the cost of this approach increases exponentially with the number of variables and is thus not optimal. Therefore, a more sophisticated search method, such as a machine learning approach, would be desirable.

3.2.2. Machine learning approach

Since chemists have already been able to synthesize thousands of MOFs, they must be able to develop a chemical intuition given the vast search space and therefore beat the brute force approach. Chemical intuition is a set of unwritten guidelines that synthetic chemists use to find the right synthesis conditions. Several papers [25, 24] have successfully shown that chemical intuition learned by a computer can find optimal MOFs faster than humans are capable of [25]. However, since machine learning is still

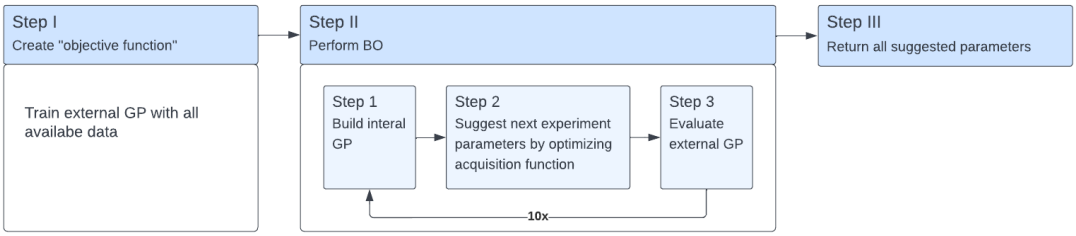


Figure 3.1.: BO workflow.

very new to the field of chemistry and is still used to a very small extent, the potential of machine learning for predicting synthesis conditions has not been explored enough and is far from being fully exhausted. Promising results have also already obtained in previous work for chemical synthesis with Bayesian reaction optimization [31].

Moreover, the parameter combinations suggested by a machine learning method may not seem intuitive to a chemist and therefore are usually left unconsidered, but could still lead to satisfactory or even better results.

3.2.3. Bayesian optimization approach

In this thesis the goal is to capture chemical intuition using BO and therefore predict synthesis parameter resulting in MOFs with a good fitness score. BO does not rely on a detailed chemical understanding of MOFs and how the fitness is computed.

In the following, first an approach for a single-objective BO with the fitness score as objective implemented with the scikit package [16] is provided, and then a multi-objective BO implemented with MOBOpt [10] is presented. The fitness is computed from various indicators measured to evaluate the performed experiment (for more details see section 3.5.6). For the implementation of BO, the calculation of the fitness is not relevant, only the value is needed. The fitness is a value between zero and one and is optimal at one.

BO requires a computable objective function to propose synthesis parameters. But since in the discussed problem statement experiments have to be performed and analyzed first to get the fitness and thus the function value of the objective function, the objective function has to be modeled to approximate the fitness function. For this purpose, GPR, hereafter referred to as external GP, was used to predict the outcome of the synthesis. It was trained with the available data and then assumed to be the reality so that multiple proposals could be produced at once. The BO workflow can be seen in Figure 3.1. The external GP should not be confused with the internal GP, which approximates the objective function and thus the external GP, and is used by

Table 3.1.: Ranges of the parameter for the all three structures.

Structure	Temperature [°C]	Time [min]	Metal [mmol]	Reactants ratio [eq L of M]	Modul. ratio [eq of M]
ZiF-8	(40.0, 145.0)	(10, 45)	(0.01, 2.00)	(0.5, 50.0)	(0.0, 10.0)
UiO-66	(80.0, 150.0)	(5, 36)	(0.01, 2.00)	(0.0, 71.5)	1.0
NH_2-UiO-66	(100.0, 175.0)	(15, 47)	(0.25, 2.00)	(30.0, 95.0)	1.0

the acquisition function to generate new sample points while minimizing function evaluations.

An alternative to the external GP would be to wait each time between suggesting parameter sets until the experiment has been performed and use the true fitness value instead of the prediction of the GP, like proposed in subsection 2.2.3. This is not practical for the MOF synthesis method used in this work, because up to six experiments can be conducted simultaneously and take the same amount of time as one experiment. The purification and analysis to determine the fitness value requires a cosiderable amount of time. Therefore, the approach with the external GP was chosen.

3.3. Dataset

The search space of the synthesis parameters is infinitely large. Therefore, it makes sense to limit the search space by restricting the limits of the parameters to chemically reasonable numbers, e.g. the reactants ratio should not be zero and the temperature should be within a range, where it is possible to obtain the specified MOF. The ranges vary depending on the MOF structure and can be seen in Table 3.1. Therefore, impossible or very unlikely condition areas are excluded based on an experts opinion.

As already mentioned in section 2.3, the focus is on three different MOF structures: ZIF-8, UiO-66, and NH_2-UiO-66. The variable parameters for the synthesis are temperature, time, metal, reactants ratio, and modulator ratio. A particular experimental condition can be described as a point in a 5-dimensional chemical space. Reactants ratio is the ratio of metal salt to linker. Metal salt is used because the metal is not added as an element but as a salt and dissolved so that it is present as an ion and combines with the linkers. The reactants ratio is fixed for UiO-66 and NH_2-UiO-66 to one. The modulator ratio is the ratio of metal and modulator.

Since the number of parameters varies depending on the structure, the reaction space has a different size depending on the structure. Exemplarily for ZIF-8, the reaction space contains more than $3.62 \cdot 10^{11}$ possible configurations.

The available learning data, consisting of 30 data points, which can be seen in Table A.6, was generated by three generations (10 data points each) of the Genetic algorithm (GA) inspired by natural selection presented in [25] with an initial dataset suggested by the grid search of SyCoFinder on the Materials Cloud website, with parameter ranges based on literature research [25].

The goal in predicting the synthesis conditions of ZIF-8 is to produce 100 nm sized particles for bioapplications in the field of drug delivery, i.e. drug incorporation and specific re-release, as in the case of recognition of a tumor receptor or similar.

In practice, a major problem is that some parameters, such as temperature, are difficult to determine accurately. Since the temperature in a room varies, it is difficult to replicate MOFs. In addition, not all measurement results are exact. Therefore, the measurements were performed several times. For example, the DLS value, the particle size component of the fitness value, is measured 15 to 20 times and the mean value is calculated as the result. Then this process is repeated and the average of the two results is calculated as the final DLS value. Therefore, it must be assumed that the measurements of the physical properties are sufficiently accurate.

3.4. Preprocessing of data

Preprocessing of the data, which in this work is normalization of all data in a uniform range is especially useful if the kernel uses some kind of location difference of the points, since the GP can then learn the correlation between synthesis parameters and the weighting of importance of the parameters without any restrictions. Without normalizing the data, the kernel is implicitly weighted due to the different ranges of the various parameters. It is highly unlikely that these weights reflect the actual correlation of the parameters. Kernels that are particularly affected by this are kernels that factor the difference between points in an arbitrary way, i.e., kernels that include a length scale. Examples of such kernels are perodic kernels like the ExpSineSquared kernel, the RBF kernel, the RationalQuadratic kernel, and the Matern kernel. In this work, normalization is done by transforming all parameters to a range of $[0, 1]$.

To highlight the difference between processed and raw data, Figure 3.2 shows the raw data on the left and the scaled data on the right in an multidimensional scaling projection (MDS) of the synthesis parameter space. A MDS projection visualizes the similarity between individuals in a dataset, computed by the pairwise Euclidean distance of the normalized variables. As can be seen, the data relationships have different shapes.

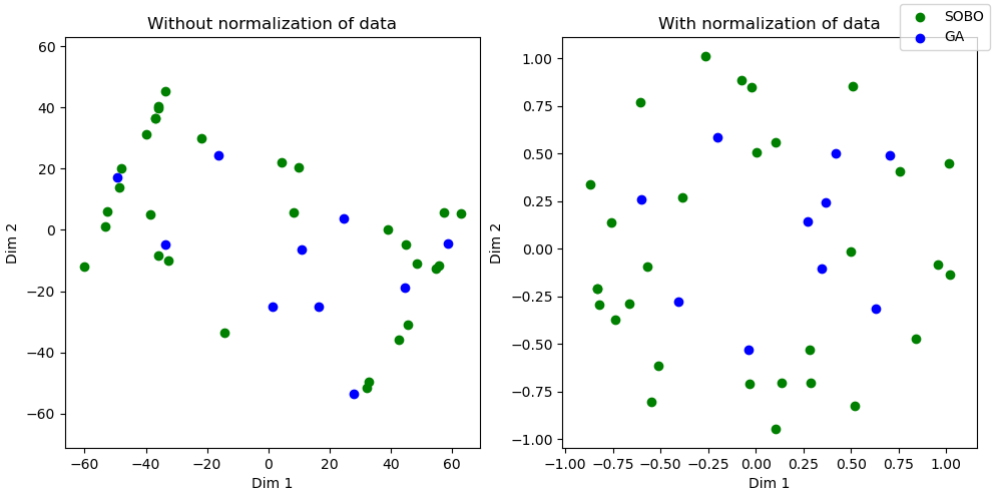


Figure 3.2.: Multidimensional scaling projection of the 5-D parameter space onto a 2-D plane. Similar conditions are close to each other. The green points are part of the given dataset from the GA and the blue ones are part of the set proposed by BO. On the left side the data is not normalized, on the right side it is.

3.5. Single-objective Bayesian optimization

To use BO, first, the best kernel of the external GP with the best hyperparameters was selected by training and evaluating the models with the available data. Then, BO could be performed, like visualized in Figure 3.1, with the intention that it converges to the optimal synthesis conditions based on the predictions of the external GP. The 10 proposed synthesis parameters were synthesized experimentally. The goal was to use the experimental data to further refine the regression model. Theoretically, the process should be repeated until the regression model fits perfectly and BO suggests the best synthesis parameters.

Due to the limitations of the project only few experiments could be performed. Thus, the focus was mainly on ZiF-8. For the others, only the first step, the basic selection of useful kernels, was performed.

3.5.1. Optimizing the kernel of the external Gaussian Process

Considering that the kernel has an immense impact on the posterior mean and variance and thus on the prediction, the choice of the kernel and its hyperparameters is crucial for the entire model. The choice of kernel and the choice of hyperparameters can be

regarded as a degree of freedom of the regression [1]. Because of the small dataset, these degrees of freedom should be kept small and therefore, not many different kernels where combined into one kernel to ensure that the number of hyperparameters is still small enough.

Not only the kernel must be selected, but also the kernel's hyperparameters, the number of which varies depending on the kernel. The influence of hyperparameters on the model were explained in subsection 2.1.5. Tuning them is important to obtain an appropriate predictive model capable of explaining the relationship between the experimental data. To highlight the importance of hyperparameter tuning, Nakayama et al. [26] showed by simulating actual material synthesis, that the worst hyperparameters display an increased number of syntheses where the global maximum can be reached with 90% probability by a factor of 2.4 compared to the optimal hyperparameters. They showed the importance of selecting hyperparameters appropriate to decrease the number of required searches. With optimal hyperparameters, BO can beat human experts already in the one-dimensional case. When optimizing several parameters, it is even more difficult for experts to find proper synthesis parameters.

Since the structure of the data and thus the structure of the objective function is not known, this cannot be exploited and does not lead to a preference for a particular kernel. Typically, kernels are chosen according to the assumptions about the structure of the data and objective function. However, since both are unknown in this setting, all kernels have been taken into account.

From a Bayesian point of view, the hyperparameters must be found that are most likely to predict the output data given the input data by the respective GP. There exist two different approaches to optimize the hyperparameters: optimizing the log marginal likelihood function of the GP and cross-validation.

Log Marginal Likelihood approach

For the first approach, the negative log marginal likelihood function gets minimized. The definition for it can be found in subsection 2.1.6. The optimal hyperparameters regarding the likelihood are obtained from the minimum of the negative log marginal likelihood function

$$\boldsymbol{\varphi} = \arg\min_{\boldsymbol{\varphi}\in\Phi} \log\, p(\mathbf{y}|X, \boldsymbol{\varphi}) \tag{3.1}$$

where $\boldsymbol{\varphi}$ is the vector of hyperparameters and Φ is the corresponding parameter space. A gradient-based optimization algorithm is used for minimization since an analytical solution is not computable. As the negative log likelihood is non-convex, the resulting optimum of the hyperparameters is not guaranteed to be optimal, since the space can have several local optima [1]. For example, gradient descent starts at some initial

position and follows the negative gradient towards a local minimum. As such, it is important to provide appropriate initial values for the hyperparameters. The advantage of the log-marginal likelihood approach is that it balances the fit and complexity of a model.

Cross-validation approach

Cross-validation splits the dataset into a training set and a test set to avoid the overfitting that occurs when testing with training data. The model would produce a perfect result because the data has already been trained with the test data, but is unable to predict useful, previously unseen data.

There is still a risk of overfitting on the test set, as the parameters can be changed until the estimator performs optimally and therefore, the model can represent knowledge about the training data and not the general data. This problem is exacerbated by the small dataset, such that results may depend on the selection of sets. Hence, the validation was performed several times with different training and test sets.

Cross-validation can be performed using different strategies, with all strategies withholding the test set for final evaluation. In this work, the performance measure for the test fold is calculated by the coefficient of determination R^2, which measures how well the model predicts the outcome and is defined by

$$R^2(\mathbf{y}, \mathbf{y}') = 1 - \frac{\sum_{i=1}^{n}(\mathbf{y}_i - \mathbf{y}')_i^2}{\sum_{i=1}^{n}(\mathbf{y}_i - \overline{\mathbf{y}}_i)^2} \tag{3.2}$$

where $\mathbf{y}$ is the real data, $\mathbf{y}'$ the predicted data, and $\overline{\mathbf{y}}$ the mean of the real data [11]. It indicates the accuracy of the observed results by the model, based on the proportion of the total variation in the observations that is explained by the model. This score is used since it shows the model fit, this means how well the predicted values match the unseen test data.

In general the cross-validation approach is compared to the log marginal likelihood approach more computationally expensive but might achieve a better representation of the dataset [1, 27].

3.5.2. Model selection of Gaussian Process Regression

To avoid overfitting, the model should not be too complicated, but detailed enough to avoid underfitting. Due to the very small dataset, there is a high risk of overfitting because the data may not be sufficient to generalize patterns across the entire data space, but only represent patterns that occur locally. Therefore, the selection of the kernel was conducted in several steps to reduce the risk of overfitting. In the following section, the division of the dataset is always performed randomly.

Table 3.2.: The best five kernels of the first selection step. The columns show the average difference of the two sets stated in the first row: internal GP (I-GP), external GP (E-GP) and true fitness (T-F).

Kernel	E-GP & I-GP	E-GP & T-F	I-GP & T-F
RationalQuadratic + ConstantKernel	0.00612	0.18070	0.18460
Matern + RationalQuadratic	0.00672	0.16604	0.16521
Matern + RBF	0.00728	0.16679	0.16564
RationalQuadratic + ExpSineSquared	0.00903	0.16765	0.16668
RationalQuadratic + RBF	0.00939	0.16723	0.16596

First selection step: Rough selection

A very rough selection of kernels was made during the first iteration. All single kernels: Constant, Dotproduct, ExpSineSquared, Matern, Rationalquadratic, RBF and White kernel, the sum of two kernels, the product of two kernels and all kernels with several exponents were tested without considering in detail the optimization of the kernel hyperparameter. This resulted in 56 different kernels. The kernels were compared based on several indicators. To compute these indicators, the available set of data points was divided into 20 training and 10 test data. From the training set, 10 points were given as initialization data to BO. The first comparison indicator assessed whether the trained external GP was able to predict the training data. Most kernels correctly predicted the test data, but some kernels were not even able to predict these training points, such as the DotProduct kernel. Kernels whose predictions of the training data deviated overly far from the actual fitness were neglected.

During the training of a GP, the hyperparameters of the kernel were optimized by maximizing the marginal likelihood using gradient-ascent. This is done by minimizing the negative log marginal likelihood. However, this is rather limited, so it is still necessary to specify suitable hyperparameters as initial parameters. Thus, each time the GP is re-trained, the hyperparameters change. The internal optimization of the parameters during the fitting process was used to propose suitable kernel hyperparameters which fit the given data. Since the log marginal likelihood may have multiple local optima, the optimizer was started repeatedly at random positions to propose good hyperparameter.

For each parameter-fitness pair in the test set, the fitness of the parameter was predicted by both the external and the internal GP and then compared to the true fitness. The difference between the internal GP and the external GP, the external GP and the true value, and the internal GP and the true value were calculated for each test point, and the average of each difference was used to evaluate each kernel. The five

best kernels were selected for the next decision step, which can be seen for ZiF-8 in Table 3.2 and for the other structures in Table A.1.

Second selection step: Evaluate kernel and its parameters

In the next step, the learning rate of the kernel was examined. The focus was on the external kernel. Again, the external kernel was trained with 20 training points, leaving 10 test points to probe the trained kernel. To provide an indicator of the learning rate, one data point was moved from the test set to the training set at each step. Hence, in the first iteration there were 20 training points and 10 test points, in the second there were 21 training points and 9 test points, in the third there were 22 training points and 8 test points and so on until there was only one test point left. Of course, only one remaining test point can lead to incorrect assumptions, which is why a final evaluation was performed afterwards. Again, the average of the difference between the true fitness and the external GP prediction at each test point was calculated and can be seen in the appendix Table A.2.

The second indicator for the comparison was the negative log marginal likelihood, explained in subsection 2.1.6. The smaller this value, the better the kernel. This likelihood was calculated when the kernel was trained with 20 data points.

The last aspect to consider in this iteration was the noise level. The noise level can either be specified directly by the GP, or considered as a hyperparameter and thus modeled as a kernel, the White kernel, as part of a sum kernel. Therefore, it can be optimized in the same way as the hyperparameters. Here, the noise was included by adding the White kernel in addition to the best kernels, and compared with the two approaches above: Learning rate comparison and log marginal likelihood. The result especially for the likelihood was reasonably clear. Kernels without noise had a likelihood around $-30,000,000$ and kernels with noise around 5. Since there is not much data available, the White kernel is not able to fit the data in this case. Thus, we are dealing with overfitting. Noise is only useful in a case with more data. The log marginal likelihood is around zero and therefore states that the model does not fit the data.

Third selection step: Final evaluation

The final evaluation attempted to address all previously neglected aspects by varying in the selection of the training and test set. In this step, the best kernels and its best hyperparameters from the previous step were compared. The first comparison was performed using 5-fold cross-validation. This cross-validation approach was used, since with other approaches the resulting subsets are too small to lead to promising results.

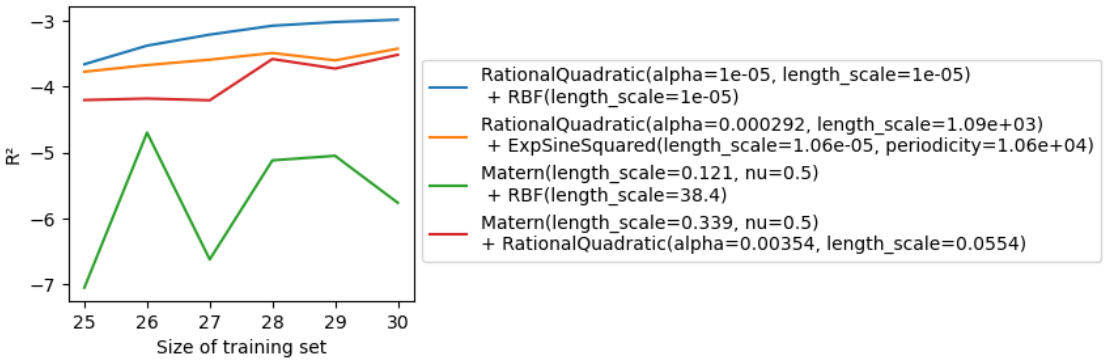


Figure 3.3.: Learning rate of the best kernels with best hyperparameters evaluated by R^2 score using cross-validation.

For each fold, R^2 and then the mean of all folds was calculated, which can be seen in Table A.3.

The second and most important decision criterion is another cross-validation approach: 20 training data points and 5 test points were randomly selected. Then the coefficient of determination of the prediction of the external GP of the test points was calculated. This process was repeated 100 times and the mean of the folds was calculated. This ensured that the score does not depend on the selected datasets and models the real score. Then the training set was increased by one and the whole process was repeated until the number of training points was 25. In this way, one can determine the learning rate of the GP, where the best kernels can be seen in Figure 3.3.

The criteria for selecting the final kernel and its parameters are the following:

1. A high learning rate, since the GP is probably not yet good enough to compute the best synthesis parameters due to the small dataset available. Therefore, it is important to have a kernel that improves drastically by just a few more training points.

2. A good overall coefficient of determination ensures already that the kernel not only has a good learning rate, but that the basic prior distribution already models the dataset well enough.

3. Some of the individual calculations of the coefficient of determination for a specific pair of training and test set were really bad. Therefore, another goal is that the number of really bad results is small.

3.5.3. Acquisition Functions

Three different acquisition functions were considered: Expected Improvement (EI), Probability of Improvement (PI), and Lower Confidence Bound (LCB). EI was chosen as the acquisition function because it provided the best results, even though the variation between acquisition functions was not significant. EI is the most commonly used acquisition function and generally provides a good balance between exploration and exploitation. For this reason, and because the results were slightly better, EI was chosen. Since there are still too many knowledge gaps in external GP anyway, the focus was more on improving the GP model than on the acquisition function, since this is the real bottleneck of the approach.

There exist also more suffisticated approaches for the acquisition function, like Knowledge Gradient or Entropy Search. These are not applied here, since the available dataset is too small, and the GP not exact enough yet to tune the acquisition function.

3.5.4. Final result of single-objective BO

The final best kernel has been found to be the RationalQuadratic + RBF kernel. The hyperparameters alpha and length scale of the RationalQuadratic kernel and the length scale of the RBF kernel are all set to 10^{-5}. The RBF kernel is a universal kernel, which means it satifies the universal approximating property [22]. A GP using a universal kernel can approximate any continuous function on a compact set with arbitrary accuracy. Therefore, it is able to learn any continuous function given enough data. However, universal kernels sometimes learn relatively slowly, which is called the curse of dimensionality. In general, the more structure considered, the less data needed, called the blessing of abstraction. Thus kernels with a significant amount of structure often require less data than flexible kernels [1, 7]. This problem is solved by summing with the RationalQuadratic kernel, since it is not a universal kernel, the learning rate is usually faster.

The RBF kernel models the long-term trend and the RationalQuadratic medium-term irregularities. A too big length scale of the RBF kernel is not suitable for predicting steeply rising peaks, which could be the global maximum. The initial variance of the RBF kernel should correspond to the changes in the physical properties. Therefore, if the variance is larger than the actual changes in the physical properties, the prediction curve is not appropriate and tends to ignore datapoints [26]. The results therefore suggest that the true objective function form steeply rising peaks.

The presented approach should lead to good predictivy accuracy, since it tries out a large number of different regression models by an basic trial and error approach. As can be seen in the tables, the scores are in general (also the one for the final result)

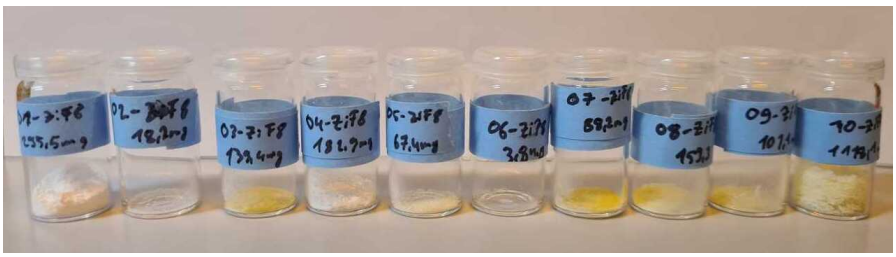


Figure 3.4.: Synthesized MOFs kindly provided by Lena Pilz, Karlsruhe Institute of Technology.

quite bad (R^2 is usually < 0) which indicates that there is still not enough training data available. Therefore, the goal is to propose the next sythesis parameters with BO to further train the GP and improve the model.

3.5.5. Realization of Bayesian optimization

To propose the synthesis parameter, the external GP was trained with all available data and then assumed to be reality. The BO was passed the available training data as intitial data to train the internal GP. The same kernel was specified for the external and internal GP. Since 10 experiments could be performed, BO ran for 10 iterations. During this process, the internal GP was trained with the new proposed data. The resulting 10 proposed parameters are the data points that the acquisition function used found most useful for further training the internal GP. This process is illustrated in Figure 3.1.

3.5.6. Evaluation of the single-objective BO approach

The synthesis parameters proposed by BO were tested experimentally and the resulting MOFs were analyzed, including the calculation of the the fitness score. This was carried out at Karlsruhe Institute of Technology by Lena Pilz and the synthesized MOFs can be seen in Figure 3.4.

Calculation of the Fitness

The fitness value consists of several measurements and is computed by

$$\text{Fitness value} = \left[1 - \left(\frac{1}{|\frac{d(g)}{d(r)-d(g)}| + 1}\right)\right] \cdot (1 - PDI) \cdot XRD. \tag{3.3}$$

XRD stands for X-Ray Diffraction. In this measurement, X-rays are diffracted at the crystal lattice or, more precisely, at the metal nodes of the nanoparticles and then,

Table 3.3.: Synthesis results of parameters proposed by BO and GA. Optimal value is one for all parameters.

SOBO				GA			
d(r)-F	PDI-F	XRD	Fitness	d(r)-F	PDI-F	XRD	Fitness
0.1293	0.00	1	0.0	0.1663	0.81	1	0.1351
0.0553	0.64	1	0.0352	0.2610	0.33	1	0.0854
0.3269	0.24	1	0.0798	0.3621	0.92	1	0.3347
0.6583	0.94	1	0.6192	0.7192	0.92	1	0.6613
0.1552	0.15	1	0.0227	0.7065	0.92	1	0.6485
0.3262	0.50	1	0.1636	-	-	0	0
0.1362	0.29	1	0.0402	0.5280	0.88	1	0.4659
0.0968	0.00	1	0.0	0.1571	0.54	1	0.0851
0.0950	0.34	1	0.0319	0.2017	0.03	1	0.0062
0.1703	0.49	1	0.0839	-	-	0	0

depending on their position in the lattice, generate a signal or, in the case of ZiF-8, several substance-specific signals, which can then be seen in the form of "peaks" in the diffractogram. The resulting pattern or positions of these peaks can then be compared to a simulation or publication to determine if the desired structure, namely ZiF-8, has emerged. XRD becomes one if this is the case, and zero if the structure is not present, which leads to a fitness of zero.

Desired are ZiF-8 nanoparticles of 100 nm size and, if possible, only one particle size should be produced, not a mixture of several. The term in round brackets consisting of $d(g)$, the goal-value, and $d(r)$, the measured real value, is responsible for the approximation to 100 nm. This term approaches one as the particle sizes approach 100 nm. The term $(1 - PDI)$ is dedicated to the criterion of achieving only one particle size if possible. PDI stands for polydispersity index and is a value between zero and one. The closer it gets to one, the greater the polydispersity, i.e. the larger the distribution of particle sizes.

Synthesis results

In the following the result of BO is compared to the second generation of the genetic algorithm (GA) [25], which already proposed the parameters for the given dataset (see section 3.3). The results can be seen in Table 3.3 and are visualized in Figure 3.5. The raw data of the proposed synthesis parameters and the measurements of the resulting MOFs can be seen in the appendix in Table A.7.

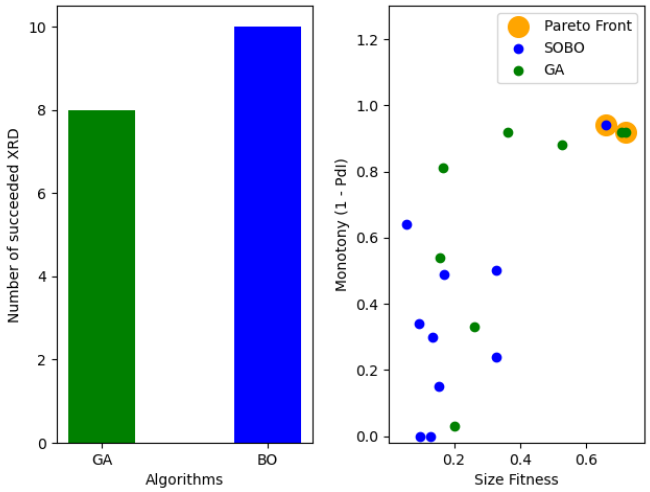


Figure 3.5.: On the left a bar plot shows the number of succeeded XRD of GA (only last generation) and BO. On the right comparison of MOFs which synthesis parameters were proposed by BO and GA in terms of their size and monotony of the particle sizes. Here, only the MOFs with succeded XRD are included.

The comparison of the XRD can be seen in the first plot of Figure 3.5. With BO, the desired structure was generated in all 10 syntheses, while with GA, even in the second learning generation, there were still two syntheses that did not lead to the desired structure. Thus, BO is clearly better concerning XRD.

The fitness values of BO are generally lower. In terms of molecular size, on the whole generation, GA is better because it has significantly more values in the closer region around the target value 100 nm than BO. However, in the absolute value of the best experiment they are similar (GA: 139nm BO: 152nm).

The BO experiments generally have much higher PDI in the whole generation than the GA. In some cases even values of one, which is the limit of the PDI. This is also reinforced by closer analyses: an accumulation (more than 5%) of at least one other particle size, i.e. not only a distribution of particle sizes (a polydisperse mixture), but even an accumulation of one other size, was clearly exhibited by one sample at BO and ambiguously by three. This was not found with GA in any of the samples. Although this information is not included in the fitness, it shows that this is more often the case with BO and not at all with GA.

The right plot of Figure 3.5 visualizes the results of BO and GA in terms of monotony and particle size. It can be seen that GA is overall better at predicting MOFs with more

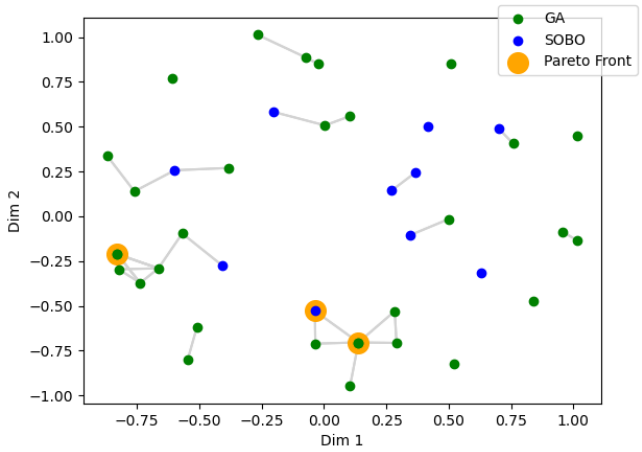


Figure 3.6.: Multidimensional scaling projection of the 5-D parameter space onto a 2-D plane. Similar conditions are close to each other and connected if they have a normalized distance below 0.25.

accurate size than BO. The best results, the Pareto front, is plotted in orange. Since both algorithms predicted one optimal synthesis parameter set, both algorithms are equally good in terms of the best proposed parameter.

Model assessment

The model is assessed by averaging the mean absolute error (MAE) in the predictions to evaluate the accuracy of the model prediction. Without the proposed ten parameters, the MAE was 15.9% and with them 14.6% when evaluated with cross-validation. Thus, an improvement can be observed. The MAE depending on the size of the training set can be seen in Figure A.2.

After finishing all interations of BO and therefore, training the internal GP, an optimal value gets suggested. However, this suggested value is most likely not good because each synthesis parameter is on the edge of the allowed range. The poor proposal is due to the few evaluations. Therefore, to obtain realistic optimal parameters, more data is needed and thus more experiments must be carried out.

In Figure 3.6, a MDS plot of the synthesis parameter space can be seen. Both the training set and the set proposed by BO are shown to visualize the similarity between individuals in a dataset. The most diverse set is located at the edges. BO does not select the most diverse set, but still suggests versatile parameters since most parameters are not connected. Considering that the variances of the synthesis parameters predicted

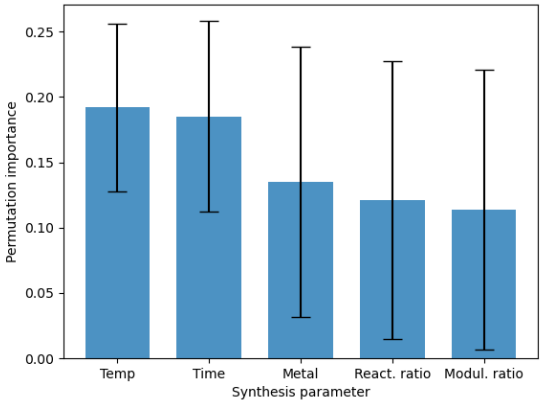


Figure 3.7.: Permutation Importance of Synthesis Parameter evaluated by permutation feature importance.

by GP are all around one, the acquisition function still performs pure exploration. 30 data points are not enough to ensure that there is enough knowledge about the entire five-dimensional parameter space.

The importance of the experimental variables was calculated by the permutation feature importance [27]. The validation performance was measured by the R^2 score and is 0.202. The importance of the synthesis parameters can be seen in Figure 3.7.

So far, the data has not been normalized like described in section 3.4. Since the prospective kernel is a sum of a RBF and a RationalQuadratic kernel, preprocessing of the data should be relevant. The R^2 score for cross-validation using the two methods described above and the log marginal likelihood of the GP give an improvement of less than 0.01% using the version with preprocessing. Also, the predictions and their variances of the GP at the points suggested by BO do not change appreciably. Although it is usually recommended to preprocess the data, it was concluded that it is not absolutely necessary with the selected kernel and its parameters, because of the small values of the hyperparameters.

When increasing the kernel's hyperparameters, the results vary drastically, this is due to the definition of the hyperparameter length scale, since it measures the correlation between synthesis parameters. Therefore, if the data would have been normalized, a different kernel may have been chosen. In addition, the preprocessing of the data leads to less variation in the results, which facilitates the comparison of kernels and hyperparameters.

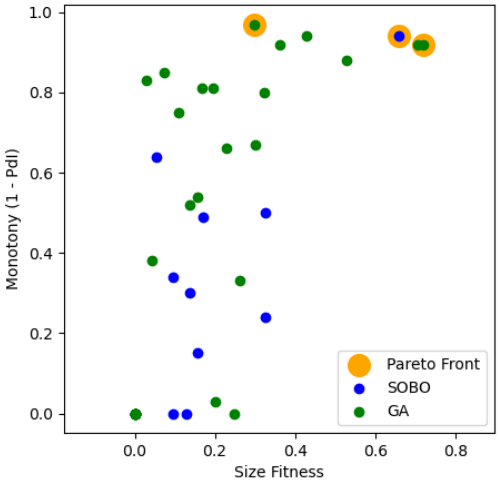


Figure 3.8.: Pareto front of dataset with parameters proposed by SOBO.

3.6. Multi-objective bayesian optimization

Since the problem of SOBO is the comparatively lower PDI and particlesize score, one solution is to include these two values directly in the BO as objectives and thus to perform MOBO on two objectives instead of only on the fitness. Thereby it could be achieved that the result is not only very good with respect to XRD, but also with respect to the other two values. Since XRD is not continuous, but can only take the discrete values zero and one, it cannot be approximated by GPR since it is a classification problem. However, since the other two objectives result in zero when XRD becomes zero, an additional XRD objective would be redundant anyway.

3.6.1. Implementation of MOBO

Since in MOBO several objectives compete with each other, as described in section 2.2.4, the Pareto set of the problem must be found, which is the set of all non-dominated points. The Pareto front with the present data is shown in Figure 3.8, where it can be seen that the values with the best fitness score are on the Pareto front. Therefore, this approach seems to be a promising method.

The MOBOpt package was used for the implementation because it computes the Pareto set with fewer objective function evaluations than other methods [10] and is implemented in a way that allowed reuse of code from the SOBO implementation. The

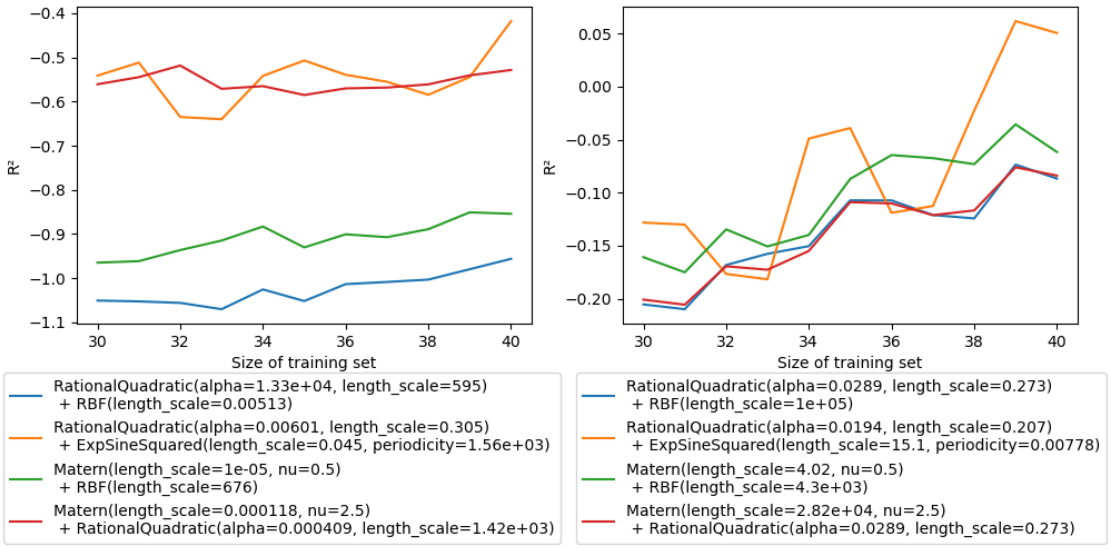


Figure 3.9.: Learning rate of 4 best kernels of MOBO evaluated by R^2 score with cross-validation. On the left the particlesize score and on the right PDI score.

data was preprocessed as explained in section 3.4. As with SOBO, external GPs were used to model the objective functions. In doing so, one GP predicted the particlesize and one PDI, using in each case not the actual value for $d(r)$ and PDI, but their fitness, which are optimal at one.

Due to lack of time, the complete selection process of the kernel and its hyperparemeter was not performed, but the best six kernels of SOBO were selected and their hyperparameters were optimized as described in subsection 3.5.2. The accuracy of the regression model was again assessed by comparing the predictions of the model with the actual outcome of experiments using among others cross-validation. Also, the proposed synthesis parameters of SOBO were used to train and validate the GPs. Each GP was trained individually.

The R^2 score of the cross-validation with increasing training data can be seen for the four best kernels with the best hyperparameters in Figure 3.9. The R^2 results are significantly better than the SOBO optimization, which is due to the prior preprocessing of the data and the 10 additional data points. Other evaluation indicators can be seen in Table A.4. Thus, the selected kernels were RationalQuadratic(alpha=0.00602, length_scale=0.304) + ExpSineSquared(length_scale=0.0451, periodicity=1.56e+03) for the particlesize GP and RationalQuadratic(alpha=0.0194, length_scale=0.207) + ExpSineSquared(length_scale=15.1, periodicity=0.00778) for the PDI GP.

With the chosen library it was only possible to specify one kernel for all internal GPs.

Table 3.4.: Synthesis results of parameters proposed by MOBO. Optimal value is one for all parameters.

	d(r)-F	PDI-F	XRD	Fitness
1	0.1166	0.08	1	0.0092
2	0.0431	0.30	1	0.0129
3	0.7372	0.90	1	0.6602
4	0.1686	0.16	1	0.0277
5	0.2426	0.71	1	0.1717
6	0.1302	0.08	1	0.0101

Therefore, for each hyperparameter the average of both GPs was chosen, since it is the best possible starting hyperparameters for all GPs and the hyperparameters are further adjusted during the execution of the MOBO.

The acquisition function used was described in section 2.2.4 and led to the selection of similar synthesis parameters as the points were located on the calculated Pareto set approximation. With the help of MOBO, an Pareto set approximation can be obtained. Five Experiments were proposed by the MOBO and one additional parameter set was computed of the mean of the Pareto set, since all decision vectors in the Pareto front were close to each other. The goal was to examine how good the calculated Pareto front already was.

3.6.2. Results of MOBO

The six proposed experiments were conducted and the results can be seen in Table 3.4 and are visualized in Figure 3.10. XRD was good for all of them as the structure always appeared. The third proposed MOF was on the Pareto front and thus optimal. Regarding the particlesize and the PDI the other predictions where generally lower than the ones predicted by SOBO and GA.

In Figure 3.11 the five predicted parameters by MOBO can be seen in terms of their predicted value, the variance and the true value. For the third MOF, which is the optimal one, the prediction was fairly precise. It can be seen that the GPs overestimated the fitness in all cases with high variance, which is the case for the second, the fourth and the fifth MOF. This leads to the conclusion that the GPs need more data to accurately represent reality. However, since only five MOFs were predicted, it is difficult to conclude something in general about the accuracy of the GPs. Nevertheless, the number of given data points is not sufficient to adequately and completely represent the five-dimensional search space.

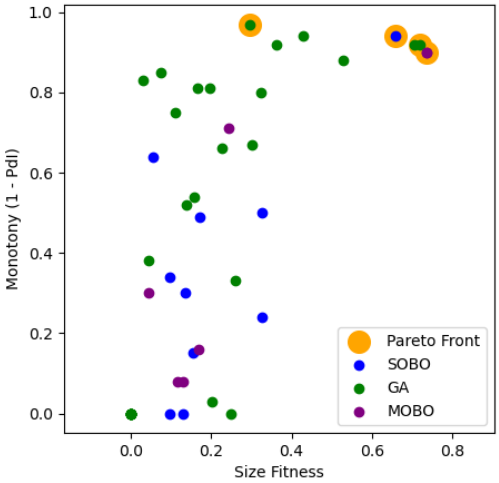


Figure 3.10.: Pareto front of dataset with parameters proposed by SOBO and MOBO.

As can be seen in Figure 3.12, the predicted parameters are very close, which is caused by the choice of the acquisition function. Three parameters are especially similar, and therefore, close to each other in the figure. Since the used library MOBOpt focuses

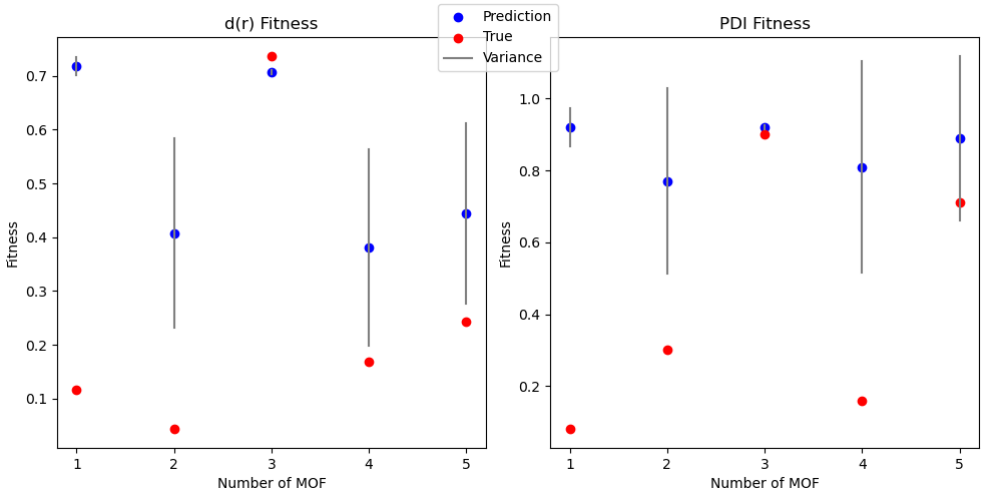


Figure 3.11.: Fitness of objectives predicted by the external GPs compared to true fitness. The corresponding numbers can be seen in Table A.5.

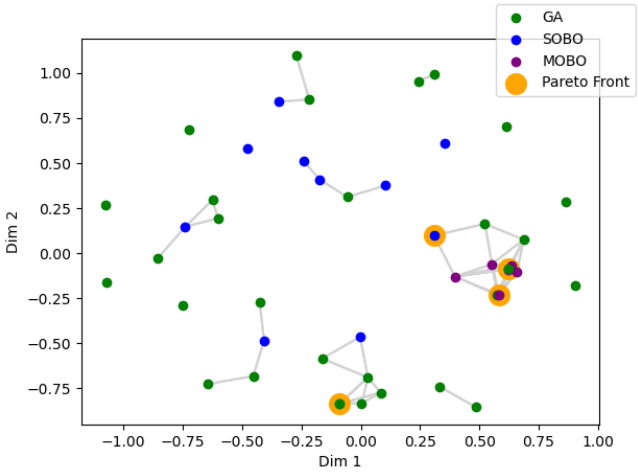


Figure 3.12.: Multidimensional scaling projection of the 5-D parameter space onto a 2-D plane of synthesis parameters proposed by GA, SOBO and MOBO. Similar conditions are close to each other and connected if they have a normalized distance below 0.25.

on approximating the Pareto set of the problem with a reduced number of function evaluations, sampling is done as close as possible to the Pareto set of the problem. This probably led to the three really close synthesis parameters being proposed, since the external GP predicts a good area there and therefore, points of this area were included in the approximated Pareto set.

4. Conclusions

In this work, Bayesian optimization with Gaussian processes was applied to find synthesis parameters for the ideal MOF composition. The MOF under study was ZiF-8. Since performing the reaction synthesis and evaluating the resulting MOF is time and resource consuming, a GP was used to model the real data distribution. The complexity of GP models scales with the number of training points. Since the used dataset was small, the GP was not complex and compuations were fast. The best kernel was selected based on cross-validation and log marginal likelihood. The hyperparameters were optimized in several steps and the most promising model was selected to perform SOBO. For ten proposed parameters, the reaction synthesis was performed and the resulting materials analysed at Karlsruhe Institute of Technology. Evaluations showed that XRD was optimal, but PDI and particlesize could be improved. Therefore, MOBO was applied to improve the result by concrete optimization of PDI and particlesize. For MOBO, six proposed syntheses were conducted. SOBO and MOBO each proposed one synthesis parameter set located on the Pareto front. In summary, it has been shown that BO can predict useful synthesis parameters already with a very small data set.

The external GP is a problematic area of implementation because it is only an approximation of reality. If the posterior is not a good fit to the available data, BO cannot predict promising synthesis conditions. Due to the small amount of available data, the selection of the best fitting kernel was difficult.

Each SOBO and MOBO has its advantages. While SOBO is faster, which is not significant because of the small dataset, MOBO is capable of optimizing several objectives simultaniously. Since a different number of experiments were performed for SOBO and MOBO, it is not possible to clearly determine the better approach. However, MOBO has significantly more optimization possibilities.

Since the selected acquisition function of MOBO resulted in similar synthesis parameter predictions, another acquisition function might lead to better results. The EI-maximin acquisition function [11] has already proven itself useful for material discovery and can obtain an adequate result with a small amount of data in a few function evaluations. Compared to other methods it is more efficient and would therefore be a good alternative for the current acquisition function.

Interesting possibilities for additional objectives of the MOBO include crystallinity and yield, which are currently not considered and also not included in the fitness. The

crystallinity is a percentage value for the X-ray diffraction pattern, i.e., it indicates the suitability of XRD.

So far, it has been assumed that the objective function can be evaluated accurately, although this is not true since the GP is an approximation to reality and every prediction has uncertainty. Therefore, an optimization approach would be to perform the Pareto front evaluation considering uncertain objectives, as described in [33].

The further overall goal is to predict the synthesis conditions for an arbitrary MOF, and not only for one structure. Therefore, further work should evaluate the kernel and hyperparameters for other MOF structures and generalize the algorithm for an arbitrary MOF structure. It would be interesting to evaluate to what extent a kernel and its parameters can be reused for other MOFs. Since their structures are very different, optimal synthesis parameters cannot be directly transferred between MOFs. Future research should investigate to what extent chemical intuition can be transferred.

A larger amount of training data would allow refinement and thus improvement of the BO model. Synthesis data from the literature could be used to obtain more data and allow more promising results to be obtained. However, there is the problem that data is usually not shared between scientists, only successful results are published, failed experiments are often neglected in publications, and the data behind them is often not available for published papers. As a result, experiments in studies are often not even reproducible. In addition, data must not only be shared, but also organized and characterized in the same way to make different datasets compatible, due to different, inconsistent metadata (information that explains and characterizes the measured or calculated data), ontologies (relationships between metadata), and workflows.

The NFDI project FAIRmat has created a FAIR (findable, accessible, interoperable and reusable) data infrastructure for condensed matter physics and solid state chemical physics. FAIRmat provides access to data and tools from and for materials synthesis, experimentation, theory, and computation [30]. Another platform for data sharing is NOMAD (Novel Materials Discovery) Laboratory, a user-driven platform for sharing and processing materials science data [6]. These projects enable efficient data sharing, better documentation, and thus new opportunities for materials science. With the help of the presented platforms, more data could be obtained and thus the GP model could be improved.

In addition, FAIRmat is currently developing new tools for fitting data, removing noise from data, and detecting and learning patterns in data. With these capabilities, an understanding of the correlation between synthesis parameters and fitness can be built and used to optimize the prior of the GP to better fit the posterior of the GP to the data.

List of Figures

List of Tables

Bibliography

[1] T. Beckers. *An Introduction to Gaussian Process Models*. 2021. DOI: 10.48550/ARXIV.2102.05497.

[2] M. Bergaoui, M. Khalfaoui, A. Awadallah-F, and S. Al-Muhtaseb. "A review of the features and applications of ZIF-8 and its derivatives for separating CO2 and isomers of C3- and C4- hydrocarbons." In: *Journal of Natural Gas Science and Engineering* 96 (2021), p. 104289. ISSN: 1875-5100. DOI: https://doi.org/10.1016/j.jngse.2021.104289.

[3] D. Bindel. *Numerical Methods for Data Science*. June 2018.

[4] P. M. Burger Benjaminand Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes, N. Rankin, B. Harris, R. S. Sprick, and A. I. Cooper. "A mobile robotic chemist." In: *Nature* 583.7815 (July 2020), pp. 237–241. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2442-2.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II." In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197. DOI: 10.1109/4235.996017.

[6] C. Draxl and M. Scheffler. "The NOMAD laboratory: from data sharing to artificial intelligence." In: *J. Phys. Mater.* 2.036001 (May 2019).

[7] D. Duvenaud. "Automatic model construction with Gaussian processes." PhD thesis. Nov. 2014.

[8] P. I. Frazier. *A Tutorial on Bayesian Optimization*. 2018. DOI: 10.48550/ARXIV.1807.02811.

[9] R. Freund, O. Zaremba, G. Arnauts, R. Ameloot, G. Skorupskii, M. Dincă, A. Bavykina, J. Gascon, A. Ejsmont, J. Goscianska, M. Kalmutzki, U. Lächelt, E. Ploetz, C. S. Diercks, and S. Wuttke. "The Current Status of MOF and COF Applications." In: *Angewandte Chemie International Edition* 60.45 (2021), pp. 23975–24001. DOI: https://doi.org/10.1002/anie.202106259. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.202106259.

[10] P. P. Galuzio, E. H. de Vasconcelos Segundo, L. dos Santos Coelho, and V. C. Mariani. "MOBOpt — multi-objective Bayesian optimization." In: *SoftwareX* 12 (2020), p. 100520. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2020.100520.

[11] A. M. Gopakumar, P. V. Balachandran, D. Xue, J. E. Gubernatis, and T. Lookman. "Multi-objective Optimization for Materials Discovery via Adaptive Design." In: *Scientific Reports* 8.1 (Feb. 2018), p. 3738. ISSN: 2045-2322. DOI: 10.1038/s41598-018-21936-3.

[12] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh. "Bayesian Optimization for Adaptive Experimental Design: A Review." In: *IEEE Access* 8 (2020), pp. 13937–13948. DOI: 10.1109/ACCESS.2020.2966228.

[13] R.-R. Griffiths and J. M. Hernández-Lobato. "Constrained Bayesian optimization for automatic chemical design using variational autoencoders." In: *Chem. Sci.* 11 (2 2020), pp. 577–586. DOI: 10.1039/C9SC04026A.

[14] F. Häse, M. Aldeghi, R. J. Hickman, L. M. Roch, and A. Aspuru-Guzik. "Gscpryffin/scp: An algorithm for Bayesian optimization of categorical variables informed by expert knowledge." In: *Applied Physics Reviews* 8.3 (Sept. 2021), p. 031406. DOI: 10.1063/5.0048164.

[15] F. Häse, L. M. Roch, C. Kreisbeck, and A. Aspuru-Guzik. "Phoenics: A Bayesian Optimizer for Chemistry." In: *ACS Central Science* 4.9 (2018). PMID: 30276246, pp. 1134–1145. DOI: 10.1021/acscentsci.8b00307. eprint: https://doi.org/10.1021/acscentsci.8b00307.

[16] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi. *scikit-optimize/scikit-optimize*. Version v0.9.0. Oct. 2021. DOI: 10.5281/zenodo.5565057.

[17] V. Hoseinpour and Z. Shariatinia. "Applications of zeolitic imidazolate framework-8 (ZIF-8) in bone tissue engineering: A review." In: *Tissue and Cell* 72 (2021), p. 101588. ISSN: 0040-8166. DOI: https://doi.org/10.1016/j.tice.2021.101588.

[18] S. L. James. "Metal-organic frameworks." In: *Chem. Soc. Rev.* 32 (5 2003), pp. 276–288. DOI: 10.1039/B200393G.

[19] L. Jiao, J. Y. R. Seow, W. S. Skinner, Z. U. Wang, and H.-L. Jiang. "Metal–organic frameworks: Structures and functional applications." In: *Materials Today* 27 (2019), pp. 43–68. ISSN: 1369-7021. DOI: https://doi.org/10.1016/j.mattod.2018.10.038.

[20] Z. Lai. "Development of ZIF-8 membranes: opportunities and challenges for commercial applications." In: *Current Opinion in Chemical Engineering* 20 (2018). Nanotechnology / Separation Engineering, pp. 78–85. ISSN: 2211-3398. DOI: https://doi.org/10.1016/j.coche.2018.03.002.

[21] Y. Luo, S. Bag, O. Zaremba, A. Cierpka, J. Andreo, S. Wuttke, P. Friederich, and M. Tsotsalas. "MOF Synthesis Prediction Enabled by Automatic Data Mining and Machine Learning**." In: *Angewandte Chemie International Edition* 61.19 (2022), e202200242. DOI: https://doi.org/10.1002/anie.202200242. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.202200242.

[22] C. Micchelli, Y. Xu, and H. Zhang. "Universal Kernels." In: *Mathematics* 7 (Dec. 2006).

[23] K. Momma and F. Izumi. "*VESTA3* for three-dimensional visualization of crystal, volumetric and morphology data." In: *Journal of Applied Crystallography* 44.6 (Dec. 2011), pp. 1272–1276. DOI: 10.1107/S0021889811038970.

[24] S. M. Moosavi, A. Chidambaram, L. Talirz, M. Haranczyk, K. Stylianou, and B. Smit. "Capturing chemical intuition in synthesis of metal-organic frameworks." In: *Nature Communications* 10 (Feb. 2019). DOI: 10.1038/s41467-019-08483-9.

[25] S. M. Moosavi, A. Chidambaram, L. Talirz, M. Haranczyk, K. C. Stylianou, and B. Smit. "Capturing chemical intuition in synthesis of metal-organic frameworks." In: *Nature Communications* 10.1 (Feb. 2019), p. 539. ISSN: 2041-1723. DOI: 10.1038/s41467-019-08483-9.

[26] R. Nakayama, R. Shimizu, T. Haga, T. Kimura, Y. Ando, S. Kobayashi, N. Yasuo, M. Sekijima, and T. Hitosugi. "Tuning of Bayesian optimization for materials synthesis: simulation of the one-dimensional case." In: *Science and Technology of Advanced Materials: Methods* 2.1 (2022), pp. 119–128. DOI: 10.1080/27660400.2022.2066489. eprint: https://doi.org/10.1080/27660400.2022.2066489.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[28] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006, pp. I–XVIII, 1–248. ISBN: 026218253X.

[29] M. Safaei, M. M. Foroughi, N. Ebrahimpoor, S. Jahani, A. Omidi, and M. Khatami. "A review on metal-organic frameworks: Synthesis and applications." In: *TrAC Trends in Analytical Chemistry* 118 (2019), pp. 401–425. ISSN: 0165-9936. DOI: https://doi.org/10.1016/j.trac.2019.06.007.

[30] M. Scheffler, M. Aeschlimann, M. Albrecht, T. Bereau, H.-J. Bungartz, C. Felser, M. Greiner, A. Groß, C. T. Koch, K. Kremer, W. E. Nagel, M. Scheidgen, C. Wöll, and C. Draxl. "FAIR data enabling new horizons for materials research." In: *Nature* 604.7907 (Apr. 2022), pp. 635–642. ISSN: 1476-4687. DOI: 10.1038/s41586-022-04501-x.

[31] 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." In: *Nature* 590.7844 (Feb. 2021), pp. 89–96. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03213-y.

[32] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. M. A. Patwary, Prabhat, and R. P. Adams. *Scalable Bayesian Optimization Using Deep Neural Networks*. 2015. DOI: 10.48550/ARXIV.1502.05700.

[33] J. Teich. "Pareto-Front Exploration with Uncertain Objectives." In: *Evolutionary Multi-Criterion Optimization*. Ed. by E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 314–328. ISBN: 978-3-540-44719-1.

[34] T. Wang, X. Wang, R. Ma, X. Li, X. Hu, F. T. S. Chan, and J. Ruan. "Random Forest-Bayesian Optimization for Product Quality Prediction With Large-Scale Dimensions in Process Industrial Cyber–Physical Systems." In: *IEEE Internet of Things Journal* 7.9 (2020), pp. 8641–8653. DOI: 10.1109/JIOT.2020.2992811.

[35] J. Winarta, B. Shan, S. M. Mcintyre, L. Ye, C. Wang, J. Liu, and B. Mu. "A Decade of UiO-66 Research: A Historic Review of Dynamic Structure, Synthesis Mechanisms, and Characterization Techniques of an Archetypal Metal–Organic Framework." In: *Crystal Growth & Design* 20.2 (2020), pp. 1347–1362. DOI: 10.1021/acs.cgd.9b00955. eprint: https://doi.org/10.1021/acs.cgd.9b00955.

[36] M. Younas, M. Rezakazemi, M. Daud, M. B. Wazir, S. Ahmad, N. Ullah, Inamuddin, and S. Ramakrishna. "Recent progress and remaining challenges in post-combustion CO2 capture using metal-organic frameworks (MOFs)." In: *Progress in Energy and Combustion Science* 80 (2020), p. 100849. ISSN: 0360-1285. DOI: https://doi.org/10.1016/j.pecs.2020.100849.

[37] H.-C. Zhou, J. R. Long, and O. M. Yaghi. "Introduction to Metal–Organic Frameworks." In: *Chemical Reviews* 112.2 (2012). PMID: 22280456, pp. 673–674. DOI: 10.1021/cr300014x. eprint: https://doi.org/10.1021/cr300014x.

A. Appendix

A.1. Single-objective Bayesian Optimization

Table A.1.: The best kernels of the first selection step of UiO-66 and NH_2-UiO-66.

UiO-66	NH_2-UiO-66
RationalQuadratic + RBF	RationalQuadratic + RBF
RationalQuadratic + ExpSineSquared	RationalQuadratic + ExpSineSquared
RationalQuadratic + ConstantKernel	RationalQuadratic + ConstantKernel
RationalQuadratic + DotProduct	RationalQuadratic + DotProduct
Matern + RBF	Matern + RBF
Matern + ConstantKernel	Matern + RationalQuadratic
Matern + ExpSineSquared	Matern + ExpSineSquared
RationalQuadratic ** 2	

Table A.2.: The best four kernels considering the learning rate evaluated by computing the average of the difference between the true fitness and the external GP prediction at each test point. Visualized in Figure A.1.

	Matern+RBF	Matern+RQ	RQ+ConstantKernel	RQ+RBF
20	0.1674	0.1672	0.1807	0.1673
21	0.1769	0.1768	0.1845	0.1769
22	0.1956	0.1952	0.1958	0.1954
23	0.1538	0.1549	0.1445	0.1535
24	0.1541	0.1531	0.1514	0.1541
25	0.1560	0.1534	0.1567	0.1554
26	0.1657	0.1642	0.1524	0.1653
27	0.1939	0.1931	0.1927	0.1926
28	0.0154	0.0439	0.0327	0.0375
29	0.0002	0.0763	0.0242	0.0714

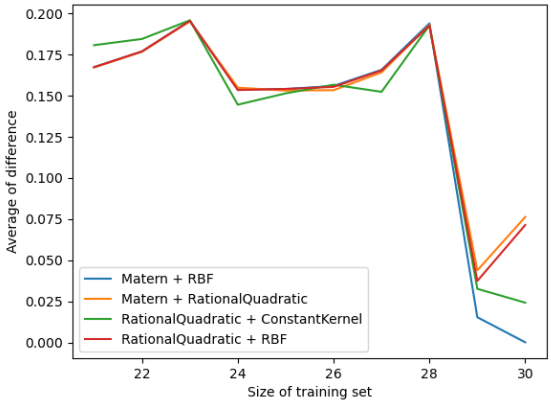


Figure A.1.: Learning rate of the best kernels computed by the average of the difference between the true fitness and the external GP prediction at each test point.

Table A.3.: The 5-fold cross-validation score of the best kernels.

Kernel	5-fold cross-validation
Matern + RationalQuadratic	-1.3706
Matern + RBF	-1.4009
RationalQuadratic + ExpSineSquared	-1.4020
RationalQuadratic + RBF	-1.4016

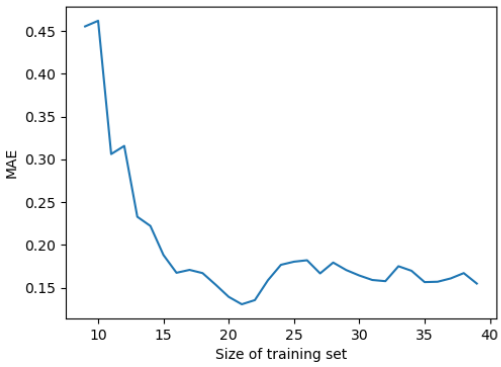


Figure A.2.: Mean absolute error depending on the size of the training set.

A.2. Multi-objective Bayesian Optimization

Table A.4.: Results of kernel optimization of MOBO. On the top half is the GP for particlesize score and on the lower half the GP for PDI. The evaluation methods are 5-fold cross-validation with R^2 score, negative log marginal likelihood, Difference of prediction of external GP and true values. Abbr.: M: Matern, RQ: RationalQuadratic, Exp: ExpSineSquared.

	M + RQ	M + RBF	RQ + Exp	RQ + RBF
5-Fold CV	-0,2946	-0,2946	-0,2702	-0,2946
N-LML	-42745332	-42745189	-42745200	-42745284
Diff GP-Test	0,1637	0,1637	0,1596	0,1637
5-Fold CV	0,1049	0,1088	0,2060	0,1047
N-LML	-2250000	-2250002	-2249999	-2250002
Diff GP-Test	0,2598	0,2351	0,2456	0,2605

Table A.5.: Fitness of objectives predicted by the external GP of the MOBO compared to the true fitness.

	Pred d(r)-F	Var d(r)-F	True d(r)-F	Pred PDI-F	Var PDI-F	True PDI-F
1	0.7176	0.0171	0.1166	0.92	0.0523	0.08
2	0.4075	0.1754	0.0431	0.77	0.2577	0.30
3	0.7064	0.0036	0.7372	0.92	0.0061	0.90
4	0.3814	0.1823	0.1686	0.81	0.2939	0.16
5	0.4436	0.1674	0.2426	0.89	0.2293	0.71

A.3. Dataset

Table A.6.: Dataset parameter, proposed by GA. Temperature, time, metal, reactantsratio, modulatorratio, fitness, XRD, d(r) in nm and PDI.

	Temp.	Time	Metal	Reactant.	Modulat.	Fitness	XRD	d(r)	PDI
1	140.00	45.00	2.00	50.00	10.00	0.2085	1	322.75	0.33
2	40.00	10.00	1.00	50.00	10.00	0.0000	0	-	-
3	40.00	45.00	2.00	0.50	5.00	0.0000	0	-	-
4	90.00	45.00	0.01	50.00	0.00	0.0000	0	-	-
5	140.00	10.00	2.00	25.25	0.00	0.0825	1	906.40	0.25
6	140.00	27.50	0.01	0.50	10.00	0.0000	0	-	-
7	40.00	27.50	2.00	50.00	0.00	0.2895	1	336.75	0.03
8	40.00	45.00	0.01	25.25	10.00	0.0000	0	-	-
9	90.00	10.00	2.00	0.50	10.00	0.2601	1	309.30	0.20
10	140.00	10.00	0.01	50.00	5.00	0.0000	0	-	-
11	40.00	14.18	0.29	25.47	10.00	0.0000	1	401.45	1.00
12	144.45	44.38	2.04	53.78	10.13	0.0710	1	729.95	0.48
13	83.88	44.73	0.77	50.00	0.00	0.1585	1	511.65	0.19
14	40.00	17.17	1.10	50.00	0.25	0.0000	0	-	-
15	140.00	10.92	0.01	19.60	8.06	0.0000	0	-	-
16	40.33	36.84	2.17	14.45	0.00	0.1505	1	440.00	0.34
17	140.00	10.00	1.53	48.31	3.75	0.0243	1	3399.00	0.17
18	63.34	29.44	1.86	9.79	9.02	0.0625	1	1368.00	0.15
19	40.00	27.50	2.00	50.00	0.00	0.4014	1	233.80	0.06
20	140.00	27.39	1.82	2.60	8.97	0.0161	1	2357.00	0.62
21	56.33	28.56	1.90	50.00	0.00	0.1351	1	601.25	0.19
22	59.71	30.37	1.95	10.16	8.00	0.0854	1	383.10	0.67
23	40.00	18.73	1.87	50.00	0.07	0.3347	1	276.20	0.08
24	130.14	11.77	1.22	48.66	2.34	0.6613	1	139.05	0.08
25	123.60	16.43	1.11	49.65	0.80	0.6485	1	141.55	0.08
26	40.00	22.19	0.39	42.13	1.80	0.0000	0	-	-
27	136.44	15.87	1.98	44.84	4.31	0.4659	1	189.40	0.12
28	51.73	41.39	0.91	23.82	0.00	0.0851	1	636.55	0.46
29	93.33	15.22	2.46	42.61	9.72	0.0062	1	495.80	0.97
30	40.32	20.97	1.63	21.02	4.34	0.0000	0	-	-

Table A.7.: Proposed synthesis parameters by SOBO (first 10) and MOBO (last 6). The order is consistent throughout the thesis to allow comparison.

	Temp.	Time	Metal	Reactant.	Modulat.	Fitness	XRD	d(r)	PDI
1	138.9	41	1.89	1.7	6.3	0.0000	1	0.1293	0.00
2	60.6	37	0.16	14.6	7.0	0.0352	1	0.0553	0.64
3	109.5	27	1.57	47.7	9.0	0.0798	1	0.3269	0.24
4	137.5	37	1.53	37.2	0.7	0.6192	1	0.6583	0.94
5	41.6	11	1.79	21.1	2.3	0.0227	1	0.1552	0.15
6	63.3	34	0.43	47.9	2.0	0.1636	1	0.3262	0.50
7	144.2	35	0.48	46.7	7.8	0.0402	1	0.1362	0.30
8	101.1	38	0.83	11.7	9.7	0.0000	1	0.0968	0.00
9	101.8	35	0.68	33.5	9.0	0.0319	1	0.0950	0.34
10	115.4	40	2.00	19.4	6.6	0.0839	1	0.1703	0.49
11	130.1	12	1.22	48.7	2.3	0.0092	1	0.1166	0.08
12	128.5	12	0.92	48.4	2.4	0.0129	1	0.0431	0.30
13	123.6	16	1.11	49.7	0.8	0.6602	1	0.7372	0.90
14	113.7	17	1.59	47.7	2.7	0.0277	1	0.1686	0.16
15	126.2	13	1.22	47.6	2.9	0.1717	1	0.2426	0.71
16	132.4	12	1.27	49.8	2.0	0.0101	1	0.1302	0.08