Accelerating the Discovery of Azobenzene-Derived Photoswitches Using Bayesian Optimization and Machine Learning

Miguel Longares Conejo (MatrNr. 399104)

Mai 22, 2025

Technische Universität Berlin
Master: Computational Engineering Science
Fakultät V

Thesis examiner:
Prof. Dr.-Ing. Merten Stender

Thesis supervisor:
Dr. Alexander Kister (BAM-Researcher)

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit eigenständig ohne Hilfe Dritter und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe. Alle Stellen die den benutzten Quellen und Hilfsmitteln unverändert oder sinngemäß entnommen sind, habe ich als solche kenntlich gemacht.

Sofern generative KI-Tools verwendet wurden, habe ich Produktnamen, Hersteller, die jeweils verwendete Softwareversion und die jeweiligen Einsatzzwecke (z.B. sprachliche Überprüfung und Verbesserung der Texte, systematische Recherche) benannt. Ich verantworte die Auswahl, die Übernahme und sämtliche Ergebnisse des von mir verwendeten KI-generierten Outputs vollumfänglich selbst.

Die Satzung zur Sicherung guter wissenschaftlicher Praxis an der TU Berlin vom 15. Februar 2023. `https://www.static.tu.berlin/fileadmin/www/10002457/K3-AMBl/Amtsblatt_2023/Amtliches_Mitteilungsblatt_Nr._16_vom_30.05.2023.pdf` habe ich zur Kenntnis genommen.

Ich erkläre weiterhin, dass ich die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Berlin, den 19.05.2025 Miguel Longares Conejo

Abstract

Photoswitches are molecules capable of reversibly changing their structure and chemical properties when exposed to specific wavelengths of light. A wide variety of photoswitches exist, each exhibiting unique behaviors, and their practical application requires precise characterization of their properties. Traditional methods for predicting these properties, such as experimental techniques and computational approaches like TD-DFT, are often time-consuming and computationally expensive. This research explores the use of Bayesian Optimization as an active search strategy to accelerate the discovery of novel molecules. The focus is on scenarios where no prior information is available, starting with zero training samples. The objective is to identify the optimal molecule with the fewest possible evaluations. While the study primarily targets photoswitches, the methodology is designed for broader applicability in molecular discovery tasks.

Photoswitches sind Moleküle, die in der Lage sind, ihre Struktur und chemischen Eigenschaften reversibel zu verändern, wenn sie bestimmten Lichtwellenlängen ausgesetzt werden. Es existiert eine große Vielfalt an Photoswitches, die jeweils ein einzigartiges Verhalten aufweisen, und für ihre praktische Anwendung ist eine präzise Charakterisierung ihrer Eigenschaften erforderlich. Herkömmliche Methoden zur Vorhersage dieser Eigenschaften – wie experimentelle Verfahren und rechnergestützte Ansätze wie TD-DFT, sind oft zeitaufwendig und rechenintensiv. Diese Arbeit untersucht den Einsatz von Bayesscher Optimierung als aktive Suchstrategie zur Beschleunigung der Entdeckung neuartiger Moleküle. Der Fokus liegt auf Szenarien, in denen keinerlei Vorwissen vorhanden ist (das Modell startet ohne Trainingsdaten) und das Ziel besteht darin, das optimale Molekül mit möglichst wenigen Evaluierungen zu identifizieren. Obwohl die Studie hauptsächlich auf Photoswitches ausgerichtet ist, ist die Methodik für eine breitere Anwendung in der molekularen Wirkstoffsuche konzipiert.

Contents

List of Acronyms

GP	Gaussian Process
AI	Artificial Intelligence
ML	Machine Learning
BO	Bayesian Optimization
DFT	Density Functional Theory
TD-DFT	Time Dependent Density Functional Theory
RBF	Radial Basis Function
CV	Cross-Validation
LOO-CV	Leave-One-Out Cross-Validation (CV)
UCB	Upper Confidence Bound
EI	Expected Improvemen
PI	Probability of Improvement
ECFP	Extended-connectivity fingerprint
SMILES	Simplified Molecular-Input Line-Entry System
SELFIES	Self-referencing Embedded String
VAE	Variational Auto Encoders
GAN	Generative Adversarial Networks
GNN	Graph Neural Network
SVM	Support Vector Machines
RF	Random Forest
RNN	Recurrent Neural Networks
SSK	Subsequence String Kernel
MSLL	Mean Standardized Log Loss
PCA	Principal Component Analysis
LSS	Least Similarity Sequence

1 Introduction

In 1956 during the Dartmouth Conference the terminology Artificial Intelligence (AI) was first introduced, proposing a research project with the claim:

> "Every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it." [McC+06]

The purpose of this project was to lay the groundwork for machines to imitate human intelligence, learning and reasoning. By that time, the power that computers had was extremely limited in comparison with today's standards. Just to realize how much has changed in that regard, we can take a look at *Moore's Law*, which predicted that the speed and capability of computers would duplicate every two years. By 2020, with the slowing of Moore's Law, computing power had increased by approximately 2^{32} times since the Dartmouth Conference in 1956, or about 4.29 billion times. Nowadays, it is fair to say that AI has reached countless fields, including banking, mobility, security, internet, science, and many more. In the context of this thesis, the application of AI will focus on the discovery of new materials, specifically targeting azobenzene-derived photoswitches.

Azobenzene photoswitches are a group of molecules with the special ability of changing their molecular structure and chemical properties as a reaction to an electromagnetic radiation, such as light exposure. The transformation from one state to the other, "acting like a switch," is most of the time reversible, meaning that a photoswitch can be turned back to its initial state by actively exposing it to a certain wavelength, through the emission of a photon or by thermal relaxation. Each state of the molecule has different properties, such as absorption spectrum, electrical conductivity, wavelength emission, switch resistance, and many other aspects[Pia19].

This individual behavior for each photoswitch derivative has drawn interest in a wide range of applications, such as: optical filters, energy storage, drug development, and even the potential replacement of traditional silicon-based transistors. This interest is driven by the need to overcome the limitations of silicon, where quantum effects and leakage currents become problematic as transistors approach atomic-scale dimensions [RH10b].

Azobenzene derivatives belong to one of the most common classes of photoswitches, undergoing reversible E-Z (trans-cis) photoisomerization. This light-induced process causes the molecule to transition between its more stable E (trans) form and the less stable Z (cis) form upon exposure to specific wavelengths [Pia19].

The molecular structure consist of two aryl rings that are connected by a diazene bridge [Mül22]. As shown in **Figure 1**, the absorption of a photon (with a ceratin wave length) triggers this isomerization mechanism, causing a change in the molec-

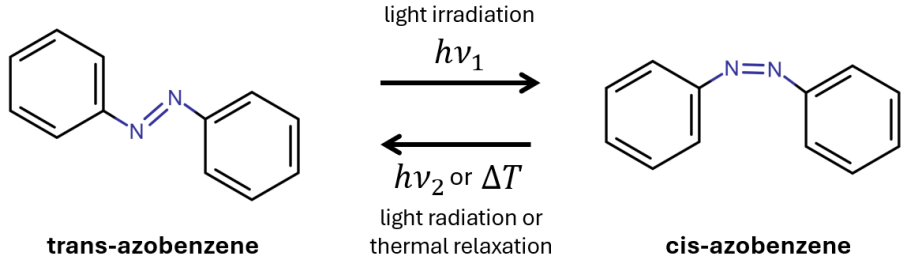


Figure 1: Azobenzene photoswitches undergo reversible structural changes triggered by a certain light irradiation ν_1, and switching back by light radiation ν_2 or thermal relaxation ΔT.

ular structure. This photochemical property is widely exploited in applications like molecular machines, drug delivery, and responsive materials.

The key aspect is that it is possible to make modifications to this structure that allow us to design photoswitches according to the required criteria, with each modification behaving differently. Each structural variation possesses unique wavelength absorption, electrical conductivity, and other properties, while retaining the ability to switch between states using electromagnetic radiation [ZZ18]. As a result, scientists have been able to use photoswitches in a variety of tasks, but the challenge is to find the optimal molecule that behaves as desired among all the possible candidates. Like any novel material developed for a scientific purpose, photoswitches must meet certain requirements to perform effectively. When considering possible azobenzene-based photoswitch models, certain rules must be taken into account. Primarily, the azo (-N=N-) bond enables the molecule to switch between states, while the two aromatic rings provide stability. Beyond this, other aspects must be considered; however, they exceed the scope of this thesis for its intended purpose [BB12].

The most basic method, as well as the most laborious, to get the isomerization properties of a molecule would be through synthesizing the molecule and experimentally characterizing it in a laboratory. However, the synthesis process can be very time-consuming due to the need for multiple reaction steps that may take several hours into consideration [Reu99] Additionally, not all azobenzene derivatives are physically synthesizable. Characterization also has some disadvantages, as individual techniques provide only partial information, requiring multiple experiments to gather enough data. For this reason, with the help of information technology, both the properties and the possibility of synthesizing the material can be predicted using computational approaches, a process known as virtual screening. The current most widely used methods are Density Functional Theory (DFT) and Time Dependent Density Functional Theory (TD-DFT), which provide information on their

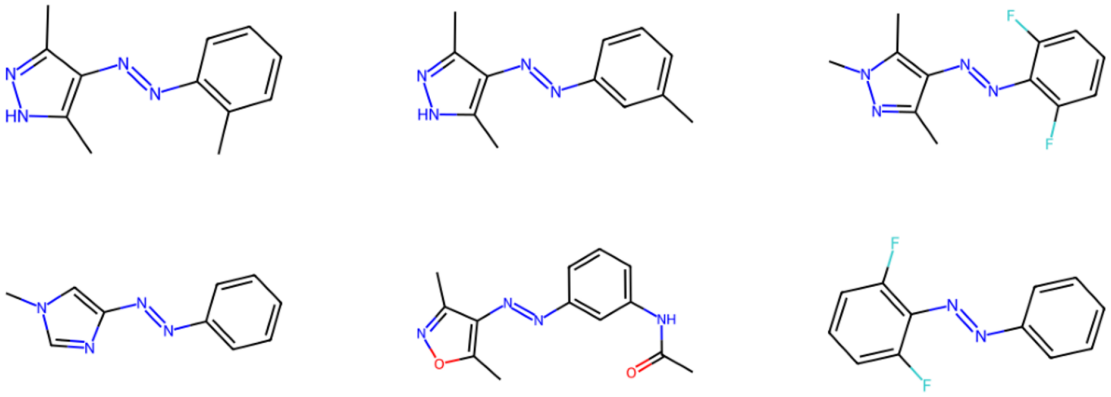


Figure 2: The photoswitch dataset contains 405 azobenzene derivatives with diverse structures designed to cover a broad chemical space. Six randomly selected molecules are shown above.

electronic structure and switching mechanisms [Wil+18]. The main advantage is that there is no need to create the material in order to test its properties, saving a lot of time. Another benefit is the ability to parallelize and scale the process when exploring large chemical spaces or testing a vast group of potential candidates, but coming in exchange of computational cost required to process large-scale simulations. Additionally, even sophisticated computational models such as DFT and TD-DFT may still reach their limits for complex molecular systems [Bur+22]. Yet the option of virtual screening has proven to be a powerful tool across many areas of materials discovery. That is why scientists combine experimental and computational methods in the search for novel molecules, using virtual screening to filter a group of potential candidates from all possible derivatives before synthesizing and characterizing them. This approach reduces the time and cost associated with traditional trial-and-error methods [Len+04]. Yet, there is significant room for improvement.

Data-driven models powered by machine learning are proving highly effective in accelerating the screening process, enabling the rapid analysis of vast datasets of molecular properties [Gri+22]. They can predict the behavior of new derivatives with high efficiency, without the computational cost associated with simulation methods such as DFT. [Lu+24]

This master thesis will focus on the implementation of machine learning algorithms to develop a data-driven model with the task of guiding the virtual screening process to determine which molecules are the best candidates. The model will employ Gaussian Processes to create a probabilistic representation of the chemical space, which is passed through Bayesian Optimization, to detect and prioritize molecules worth further investigation. The model will be evaluated by identifying the molecule

with the highest wavelength emission during state transitions within a database of photo-switches. This approach simulates a real-world research scenario, where scientists must efficiently screen a pool of unknown candidates to find those with the most desirable properties.

The next section will provide an overview of the mathematical foundations of Gaussian Processes and Bayesian Optimization to offer a general understanding of the functionality and application of these algorithms. This knowledge will serve as a foundation for later discussions on how to customize the model and its parameters to improve performance. Section 3, State of the Art, will present the latest studies related to this field, emphasizing the key challenges associated with the creation of data-driven models for chemical applications. Taking sections 2 and 3 as a starting point, an approach to building a suitable model will be presented, examining how different configurations influence the performance of the algorithm. Once a fitting framework is defined, including possible model-variations, in chapter 5 different experiments and tests will be performed to analyze how well the model generalizes across different tasks and datasets. The intention of this research is not only to predict properties of photo-switches but also to expand the model's usefulness to other molecules and materials. Finally, the thesis will conclude with a discussion and conclusion, revealing strengths and limitations of the approach.

In a nutshell, this work aims to contribute to the development of data-driven strategies for accelerating molecular screening and optimization.

2 Background

In this chapter, key concepts and theories will be explained to provide an understanding of the principal algorithms, presenting the essential mathematical principles of Gaussian processes and Bayesian optimization and revealing various configurations for distinct tasks. This will be of great help in later understanding the transition between different methods within the model, as well as exploring new techniques to overcome future challenges.

2.1 Gaussian Processes

The principal tasks of machine learning are to make predictions, identify patterns or take decisions from a dataset. Various methods have been developed, varying from simple linear regressions to more complex structures such as neural networks, each of them having its strengths and weaknesses. Gaussian Process (GP) is a powerful technique that excels in situations where the available data to train the model is scarce, capable of learning complex data patterns without being restricted by a predetermined form [Mac+98]. GP is a supervised and non-parametric approach,

meaning it does not assume a specific model structure or set of parameters in advance, making it ideal for regression and classification tasks. Predictions from GP provide an expected prediction value accompanied by a measure of uncertainty, making it very practical for situations where understanding the doubt in a prediction is essential. An intuitive example would be to predict the price of a house based on its size in square meters. In this case, a simple linear regression would suffice to make predictions, since larger houses tend to be more expensive. GP would be perfectly capable of fulfilling the task while also providing information on the price variance for houses of the same size. However, for less intuitive cases where linear regression would not suffice, GPs are truly efficient in adapting to the underlying structure of the data. They can model objective functions influenced by multiple factors, including some that may not be directly observable, making GPs an ideal choice for more challenging prediction tasks.

GP are probabilistic models that can be viewed as an extension of high-dimensional multivariate Gaussian distributions. A GP is fully specified by a **mean function** $m\colon \mathcal{X} \to \mathbb{R}$ and a **covariance function** $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (kernel), which can be defined as follows [WR06]:

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

where:

- $m(\mathbf{x})$ is the mean function, given by $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$,
- $k(\mathbf{x}, \mathbf{x}')$ is the covariance function, defined as $k(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}(f(\mathbf{x}), f(\mathbf{x}'))$.

Meaning that for any finite set of inputs $\{x_1, x_2, \ldots, x_n\}$ in the input space $\mathcal{X}$, the corresponding function values $\{f(x_1), f(x_2), \ldots, f(x_n)\}$ follow a multivariate Gaussian distribution:

$$\mathbf{f} = [f(x_1), f(x_2), \ldots, f(x_n)]^\top \sim \mathcal{N}(\mathbf{m}, \mathbf{K}),$$

where:

- $\mathbf{m} = [m(x_1), m(x_2), \ldots, m(x_n)]^\top$ is the mean vector,
- $\mathbf{K}$ is the covariance matrix with entries $K_{ij} = k(x_i, x_j)$.

A Gaussian process can be interpreted as a distribution over functions. For every input $x \in \mathcal{X}$, the corresponding output $y(x)$ is not a fixed value but instead follows a Gaussian distribution characterized by a mean $m(x) = \mu(x)$ and variance $k(x, x) = \sigma^2(x)$. Specifically:

$$y(x) \sim \mathcal{N}(\mu(x), \sigma^2(x)).$$

2.1.1 Principal of Gaussian Process

A GP is a method for making predictions based on past data. First, it establishes an idea of how different input points and their corresponding labels might be connected. In the example of predicting house prices, it represents assumptions about how house sizes and prices might be related before seeing any data; called the prior. Then, by incorporating the observed data like, known house prices, the model updates these assumptions to form the posterior, allowing for improved predictions. This updated (trained) model not only predicts the values but also provides a measure of confidence in its predictions.

In Gaussian Processes, the marginal and conditional properties of multivariate Gaussian distributions are the cornerstone for making predictions through the learned 'posterior' [WR06]. Here, $\mathbf{X}$ typically denotes the collection of input points in a dataset, with the shape $(n_{samples} \times m_{features})$ and $\mathbf{y}$ represents the corresponding target values for each sample. For instance, to predict new samples $\mathbf{y}_2 = f(\mathbf{X}_2)$ based on observed data $(\mathbf{X}_1, \mathbf{y}_1)$, we start with the fact that $\mathbf{y}_1$ and $\mathbf{y}_2$ are jointly Gaussian, as they are derived from the same multivariate Gaussian process, the joint Gaussian distribution can be written as:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right),$$

where:

- $\boldsymbol{\mu}_1 = m(\mathbf{X}_1)$, $\boldsymbol{\mu}_2 = m(\mathbf{X}_2)$,
- $\boldsymbol{\Sigma}_{11} = k(\mathbf{X}_1, \mathbf{X}_1)$,
- $\boldsymbol{\Sigma}_{22} = k(\mathbf{X}_2, \mathbf{X}_2)$,
- $\boldsymbol{\Sigma}_{12} = k(\mathbf{X}_1, \mathbf{X}_2)$, and $\boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{12}^\top$.

From this joint distribution, as shown in [Sta22], the conditional distribution of $\mathbf{y}_2$ given $\mathbf{y}_1$, $\mathbf{X}_1$, and $\mathbf{X}_2$ is obtained as:

$$p(\mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{X}_1, \mathbf{X}_2) = \mathcal{N}(\boldsymbol{\mu}_{2|1}, \boldsymbol{\Sigma}_{2|1}),$$

where:

$$\boldsymbol{\mu}_{2|1} = \boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1),$$
$$\boldsymbol{\Sigma}_{2|1} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}.$$

If we assume a zero-mean prior, the conditional mean and covariance simplify to:

$$\boldsymbol{\mu}_{2|1} = \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\mathbf{y}_1, \quad \boldsymbol{\Sigma}_{2|1} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}.$$

The predicted mean $\boldsymbol{\mu}_{2|1}$ represents the expected value for the new samples given the observed data $(\mathbf{X}_1, \mathbf{y}_1)$, while the diagonal elements of the covariance matrix $\boldsymbol{\Sigma}_{2|1}$ capture the variance of the predictions, reflecting the uncertainty at each of them.

A key aspect to take into considerations, is the dimensionality of the joint Gaussian distribution in a GP, which is defined by the number of points considered n in the input space $\mathcal{X}$. In practice, this means that the larger the training dataset is, the larger the covariance matrix $\mathbf{K}$ becomes, making the computational complexity scale as $\mathcal{O}(n^3)$ due to the required matrix inversion of $\boldsymbol{\Sigma}_{11}^{-1}$[WR06].

2.1.2 Covariance functions – Kernels

Covariance functions in the context of GP, also called kernels $k(\mathbf{x}, \mathbf{x}')$, are one of the most important aspects when defining the model, as they describe assumptions on how the objective function behaves, controlling the smoothness, local behavior, variance, and so on. Kernels' main task is to describe how points in the input space are close to each other, making them have similar output values. Given a set of points $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ in the input space, the kernel function creates a Gram matrix K describing how all the points are related to each other, formed as:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j).$$

There are many ways to describe similarity between points using covariance functions, and each of them describes point similarities in distinct ways. Therefore, this section provides an overview of the most commonly used kernels, as well as some concrete ones that have been defined particularly to identify resemblance between molecules.

Squared Exponencial Kernel:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2l^2}\|\mathbf{x} - \mathbf{x}'\|^2\right)$$

Also known as the Radial Basis Function (RBF) kernel, considered one of the default choices for GP kernels [WR06]. The covariance function possess two parameters and since the function is infinitely differentiable it generates infinitely differentiable smooth functions. The lengthscale l defines the 'smoothness width area'. The larger the value of l, the smoother the generated functions are. In contrast, if l is chosen to be small, the resulting functions will be more abrupt. The output variance, σ^2, determines the average distance of the functions from their mean, similar to setting the amplitude.

Matérn Class Kernel:

Stein [Ste99] pointed out that assuming functions are extremely smooth is often not appropriate for many physical systems and suggested using the Matérn class instead [WR06]. The Matérn family is defined by:

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|x - x'\|}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}\|x - x'\|}{\ell} \right)$$

where K_ν is a modified Bessel function, Γ the gamma function and ν,ℓ are positive parameters. ℓ defines like in the RBF how far the influence between points extends and ν controlls how smooth the function is. In most cases, the smoothness parameter is chosen to be a half-integer value (like $\nu = \frac{3}{2}$ or $\nu = \frac{5}{2}$), which simplifies the expression and makes the covariance depend only on the distance between two points. This flexibility makes the Matérn kernel particularly useful for modeling data that may not behave in a perfectly smooth way [Gar23].

Dot-Product Kernels:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_0^2 + \mathbf{x} \cdot \mathbf{x}'$$

A family of non-stationary kernels defined by the inner product of the input vectors x and x'. The effect of this kernel is equal to simply doing a Bayesian linear regression, meaning that it would be better to directly approach the task with a Bayesian linear regression, which scales $\mathcal{O}(n)$ clearly better in comparison to GP $\mathcal{O}(n^3)$ [Duv14]. Although the dot-product kernel typically doesn't offer the flexibility required for many regression tasks, it is useful to mention for two reasons. First, the polynomial kernel defined as $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d$, which is a generalization of the dot-product kernel can be effective in high-dimensional classification problems, especially when x is a binary vector [WR06]. Second, the dot-product kernel is often combined with other kernels to introduce linearity into the prior, which can be advantageous in various modeling tasks [Duv14].

Tanimoto Kernel:

$$k(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x} \cdot \mathbf{x}'}{\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - \mathbf{x} \cdot \mathbf{x}'}$$

The Tanimoto kernel certainly doesn't belong to the most common used kernels for general GP tasks, but in context of this research it holds a significant value for molecular applications. The tanimoto coefficient identifies molecular structures that are closely related to each other and has become the standard to measure similarity between compounds [FVB02]. It is worth mentioning that it applies for a certain

molecular representation in form of binary vectors, called chemical fingerprints. How fingerprints and other representation methods encode atom types and bonds will be explained in section 3. The important aspect to mention is that the tanimoto kernel operates similar to the dot product (also non-stationary), where $\mathbf{x} \cdot \mathbf{x}'$ represents the number of shared features, while $\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - \mathbf{x} \cdot \mathbf{x}'$ stands for the number of non-shared properties, which penalizes the dissimilarity between two molecules.

There are many more kernels that can be applied in a Gaussian Process to suit different tasks. Additionally, it is possible to create custom kernels by combining existing covariance functions through multiplication and summation. However, in order to obtain a valid covariance matrix, it is crucial that the chosen covariance functions satisfy two key conditions to ensure they are suitable for use in a Gaussian Process [Cut09]:

1. The covariance function $k(x_i, x_j)$ must be symmetric, resulting into a symmetric Gram matrix K.

$$k(x_i, x_j) = k(x_j, x_i) \quad \forall x_i, x_j \in X.$$

2. The covariance function must be positive definite, to ensure that the Gram matrix K is positive semi-definite. For any set of real coefficients $c_1, c_2, \ldots, c_n$, the following condition holds:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \quad \forall n \in \mathbb{N}, \quad x_1, \ldots, x_n \in X, \quad c_1, \ldots, c_n \in \mathbb{R}.$$

Meaning that the Gram matrix K has non-negative eigenvalues:

$$\lambda_i \geq 0, \quad \forall i.$$

2.1.3 Model Selection

The concept of model selection for GP is referred to the search for an optimal kernel and calibration of its hyperparameters to archive the best possible prediction performance [WR06]. Like in other Machine Learning (ML) algorithms, this method is the process of training the model so that it fits the training data as good as possible.

Hyperparameter Optimization: Given a fixed kernel, tuning the hyperparameters to fit the observations is essential for later predictions. There are two common approaches when optimizing the GP model:

One is the bayesian approach, offering one effective solution for the hyperparameters, which involves maximizing the marginal likelihood, also known as the evidence, given by:

$$\theta^* = \arg\max_{\theta} p(y \mid X, \theta)$$

$$p(\mathbf{y} \mid \mathbf{X}, \theta) = \mathcal{N}(\mathbf{0}, K_\theta + \sigma^2 I)$$

where K_θ is the Gram matrix set up by the parameters θ, and $\sigma^2 I$ represents the noise in the observations. The optimal parameters are found by maximizing the log marginal likelihood:

$$\log p(\mathbf{y} \mid \mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^\top (K_\theta + \sigma^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log |K_\theta + \sigma^2 I| - \frac{n}{2}\log 2\pi.$$

The equation has three terms: the first describes how well observations y fit with K_θ and $\sigma^2 I$, while the second term independent of y represents the complexity of the model. The third term is a normalizing factor.[WR06]

Another method for the process of model selection is through CV. By splitting the labeled data into a training set and a validation set, this approach minimizes the generalization error by finding the optimal parameters. As mentioned earlier, a common use case of GP is for training tasks with limited data, which is why CV is most of the time performed with a k-fold validation setting. In extreme cases, the number of folds is equal to the number of training samples, known as Leave-One-Out CV (LOO-CV), but the computational cost of training n models is, in most cases, excessive. As for the loss function, the most commonly used is the squared error loss, yet any other loss functions are adequate for the task.

In this thesis, the marginal likelihood will generally be used for parameter optimization. However, in special cases where it might not yield appropriate results, LOO-CV will be employed as a useful tool to cross-check the results.

2.2 Bayesian Optimization

Up to this point, it has been stated that GP is capable of approximating the true objective function, such as predicting the E isomer transition wavelength for a dataset of photoswitches. However, as mentioned in the introduction, researchers are interested in identifying the best candidates from a group of considered compounds, and this is where Bayesian Optimization (BO) becomes relevant.

BO requires a reliable model of the system being studied, which can describe the possible behaviors of the objective function. The reason GP is particularly useful in

this context is that it provides a statistical model that not only predicts the values of the objective function but also quantifies the uncertainty in these predictions. The accuracy of this surrogate model, built from the observed data, is critical for the optimization process, as it directly impacts the performance and effectiveness of BO, leading to better results [Gar23].

BO is a sequential, model-guided algorithm, that strategically searches for a point x^* in the input domain corresponding to the maximal value in the real valued objective function f^*:

$$x^* \in \arg\max_{x \in X} f(x); \quad f^* = \max_{x \in X} f(x) = f(x^*).$$

A major difference between other optimizers, resides on the advantage that the algorithm does not require an explicit expression for f, enabling the option to optimize black box systems. Therefore BO has proven effective in optimizing objectives where functions [Gar23]:

- are expensive to evaluate
- don't posses a meaningful expression behaving as "black boxes"
- gradient difficult of impossible to compute

The BO framework, depends on two key components: A statistical surrogate model provided by GP and an acquisition function $\alpha()$ functioning as a policy guiding the algorithm to select the next best candidate to find the optimal point x^*.

2.2.1 Acquisition Functions

In BO, **acquisition functions** determine the series of points to measure, so that after a number of iterations the best possible point x^* is found. These policies are designed to consider both the expected mean and the uncertainty provided by the GP. By maximizing (or minimizing, depending on the objective), the acquisition function selects the most prominent candidate point for evaluation at each iteration.

Common Acquisition Functions

Here are some of the most common Acquisition functions applied in BO, each with its own approach on how to take the mean and variance into consideration:

1. **Upper Confidence Bound (UCB):** UCB is the simplest acquisition function to observe how predicted mean and uncertainty are combined to determine a value for each point. It balances exploration and exploitation by adjusting the parameter λ.

$$UCB(x) = \mu(x) + \lambda\sigma(x)$$

2. **Probability of Improvement (PI):** PI quantifies the probability of improving the best observed value over the search space. Φ is the cumulative density function, and ξ is the parameter that encourages exploration – the higher the parameter ξ is set the more it will tend to explore new regions. To gain an intuition on how it works, if none of the predictions $f(x) \triangleq \mu(x)$ provided by GP can best the f_{best} value, the nominator will become negative for all considered points, meaning that a higher uncertainty $\sigma(x)$ will favor equal valued nominators.

$$PI(x) = \Phi\left(\frac{\mu(x) - f_{\text{best}} - \xi}{\sigma(x)}\right)$$

3. **Expected Improvemen (EI):** PI considers the likelihood of a given point to surpass the best observed value, but doesn't take into consideration how much improvement will be achieved. EI measures the expected improvement over the best observed value and is perhaps one of the most used acquisition functions for sequential maximization. It encourages exploration in uncertain regions but focuses on areas with the highest expected gains.

$$EI(x) = \mathbb{E}\left[\max(0, f(x) - f_{\text{best}})\right]$$

$$EI(x;\xi) = (\mu(x) - f_{\text{best}} - \xi)\Phi\left(\frac{\mu(x) - f_{\text{best}} - \xi}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\mu(x) - f_{\text{best}} - \xi}{\sigma(x)}\right)$$

Exploration vs. Exploitation Trade-off

The efficiency on how BO with the acquisition function finds the optimal point x^*, relies on how the balance between exploration (try uncertain areas) and exploitation (try known promising areas) are managed. Parameters like λ and ξ enable the adjustment of this balance to improve the optimization process. In the case of the UCB acquisition function, if λ is set to a high value, the uncertainty plays a larger role, prioritizing exploration over exploitation.

To conclude, acquisition functions are essential for the BO process to reduce the number of function evaluations needed to extract the optimum value. Naturally, as mentioned before, the accuracy of the statistical model provided by GP is crucial; otherwise, no matter which acquisition function is used, it would be like groping in the dark.

2.2.2 Illustrative example of Bayesian Optimization

GP task is to illustrate a statistical model of the high-dimensional chemical space, passing information about its predictions as well as variations to the BO. The idea

of this research is to find the best possible photoswitch derivative with the highest transition wavelength.

To get familiar with these processes before entering into the subject, consider a one-dimensional GP where, in each iteration, BO computes its belief about the function and selects the next point to evaluate using the defined acquisition function, in search of the optimum. The following example displays how the framework improves its predictions step by step in **Figure 3**.

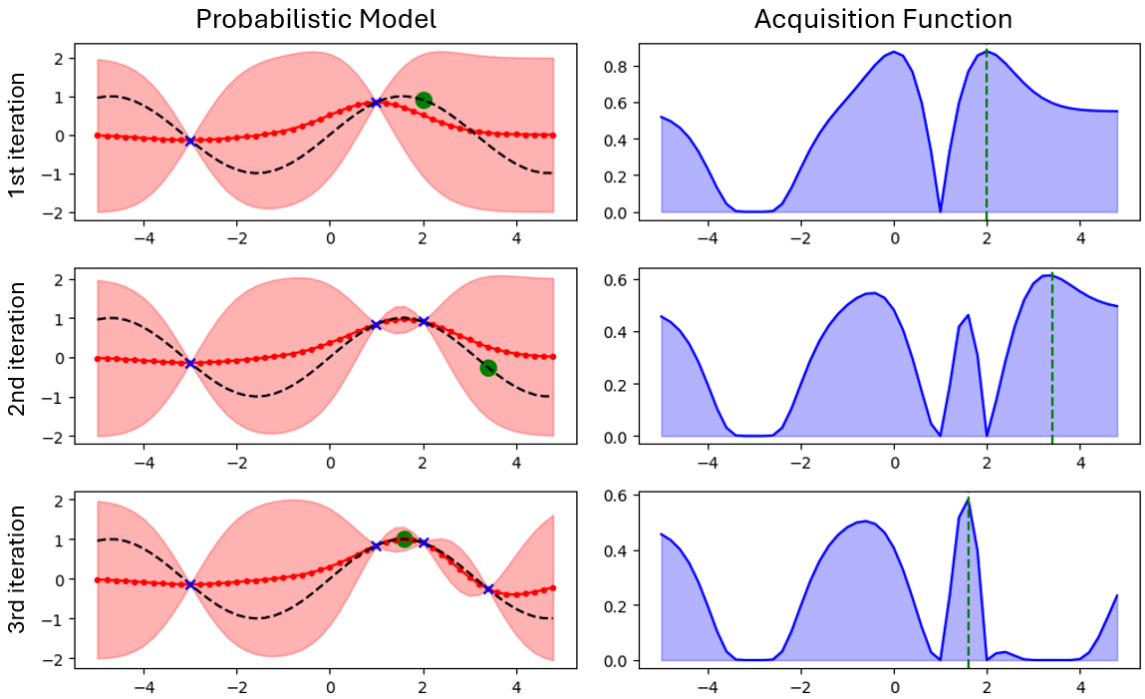


Figure 3: A simple example of maximizing a sinusoidal function using BO with two initial reference points and three iterations. Left: GP posterior estimates. Right: corresponding acquisition functions.

On the left side the probabilistic models generated by the GP are shown. The red dotted line represents the prediction mean $\mu(x)$ provided alongside with the uncertainty/variance illustrated by the light red shaded region. In the first iteration the model gets trained with two samples marked as blue crosses and the black dashed line stands for the true objective function from which we are trying to obtain the optimum value.

After obtaining the value for the selected point, the GP will be trained again, taking the new point acquired into consideration. This process is considered an iteration, where the GP is fitted on the seen data, BO picks an evaluation point, and it gets included into the training data for the next iteration.

Each row in **Figure 3** corresponds to an iteration. In the first and third iterations, it is observable how the BO algorithm exhibits an exploration behavior by selecting points near the highest observed value, while in the second iteration, the algorithm chooses to pick a point from an unknown region with a high value of uncertainty, expressing an exploration behavior.

In the context of this work, the use of GP (Gaussian Processes) and BO (Bayesian Optimization) will differ from the straightforward example provided above. Instead of a one-dimensional case, multiple dimensions will be involved, making the search space significantly more complex. Additionally, since the domain space is discrete, the objective function will not be continuous and will lack a smooth or easily interpretable structure. Rather, it will behave like a black box, with its characteristics largely unknown and accessible only through direct experimentation or observation.

3 State of the art

In this section a deeper insight into the current research will be provided, starting by pointing out the most important conclusions derived from the paper *"Data-driven discovery of molecular photoswitches with multioutput Gaussian processes"* [Gri+22], which serves as a foundation for this master thesis. Following this, we introduce the broader field of ML in cheminformatics, focusing on the most widely used molecular representations and a review of recent notable studies that demonstrate the application of ML models in this domain.

3.1 Discovery of photoswitches with Gaussian Processes

The study from the Chemical Science Journal *"Data-driven discovery of molecular photoswitches with multioutput Gaussian processes"* [Gri+22] focuses on exploring the chemical space of azobenzene photoswitches using AI, with the goal of predicting the suitability of various compounds for specific applications. Three components were carefully studied to build the prediction pipeline: Dataset, ML-model and a molecule representation.

Starting with the dataset, a total of 405 photoswitches azobenzene derivatives, with a wide variety of substitution patterns, were experimentally labeled with multiple properties such as: Rate of thermal isometration, irradiation wavelength, transition wavelengths (experimental and DFT-computed) and many more. The primary objective was using the model to predict transition wavelengths.

Regarding model selection, multiple models were evaluated: GP, random forests, graph convolutional networks, Bayesian neural networks and LSTMs. Final results provided GP using the tanimoto kernel to be the most accurate in predictions, in

addition to the advantage of providing an uncertainty estimate (Details on the ran experiments and evaluations are found in the Supplementary information file).

Having the key components defined for the prediction pipeline, a comparison against two of the most utilized TD-DFT methods (CAM-B3LYP, PBE0) was performed. Training the GP with the leave-one-out validation, the model was able to outperform the PBE0 method and provide a similar performance to the CAM-B3LYP. Regarding time performance, the ML-Model provided results in less than a minute while the TD-DFT methods needed over 200 days – In terms of runtime, there is no contest.

An interesting direction for future research, highlighted at the conclusion of this study and serving as the primary focus of this thesis, is the integration of BO to guide the molecular search process.

3.2 Molecule Representation

Molecules can be described in many ways: Molecular formulas, which most people are familiar with, indicate only the elements present in a compound, such as H_2O for water. However the formula does not give implicit information on how the atoms are bond together. Being aware of a compounds structure is important to make predictions about how it will behave chemically and physically. As previously seen at the introduction section [**Figure 1**] molecules can be described on a 2D sketch by a graph, but there are other methods to represent the structure of molecules also from a 3D perspective [Lib25].

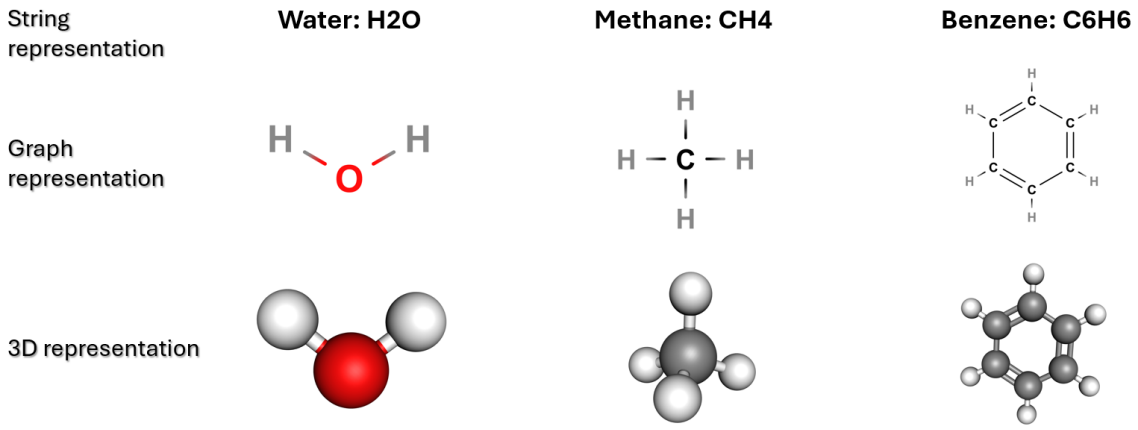


Figure 4: Examples of different molecular representations: string, graph, and 3D, for water (H_2O), methane (CH_4), and benzene (C_6H_6). Adapted from [Lib25].

With the introduction of bio-cheminformatics and the usage of AI, the need

emerged to represent molecules in a way that machines were capable to understand. Numerous methods have been developed to achieve this goal, yet there is still the issue that many molecular representations are irregular, meaning that different molecules are mapped to the same representation and vice versa [WGL22]. A perfect general representation does not exist; for each circumstance there is a representation me thod that is more appropriate, considering multiple factors like ML-model, molecular structure and global objective. For this reason, an overview of possible machine-readable representations will be presented.

3.2.1 Vector representation

The most obvious advantage of vectors is their inherent compatibility with similarity kernels, making them well-suited for a wide range of machine learning models in chemistry [WGL22]. Another aspect is the flexibility they provide when models have to be changed or customized, making the usage of vector representations very attractive at first hand. In this subsequent section, a variety of different vector representations for molecular structures will be presented.

Molecular Fingerprints

The Extended-connectivity fingerprint (ECFP) [RH10a] is probably one of the most widely used vector representations for machine learning in chemistry, largely due to the availability of open-source tools such as RDKit, which can efficiently generate these so-called 'Morgan fingerprints' as large binary vectors [WGL22]. Multiple studies show that the usage of ECFP representations consistently performs very well for different ML-Models such as Random Forest, GP and Bayesian Neural Networks [Gos22] [Tom+23]. The generation of ECFP involves the localization of certain molecule properties (eg. number of bonds, charge, number of hydrogens, etc...) which are transformed into hashes. This process is repeated iteratively, each time considering a larger radius of neighboring atoms, as illustrated in **Figure 5**. The final result is a fixed-length binary vector representation of the molecule, typically of size 2048.

Fragment representation

Fragment representation describes molecules by counting the occurrences of explicitly selected substructures (fragments), resulting into a vector of the same size as the number of considered substructures[Bas08]. Unlike ECFP, the elements of the resulting vector are positive integers and the vector dimension is typically smaller, provided that the number of considered substructures is kept low. There exists a wide variety

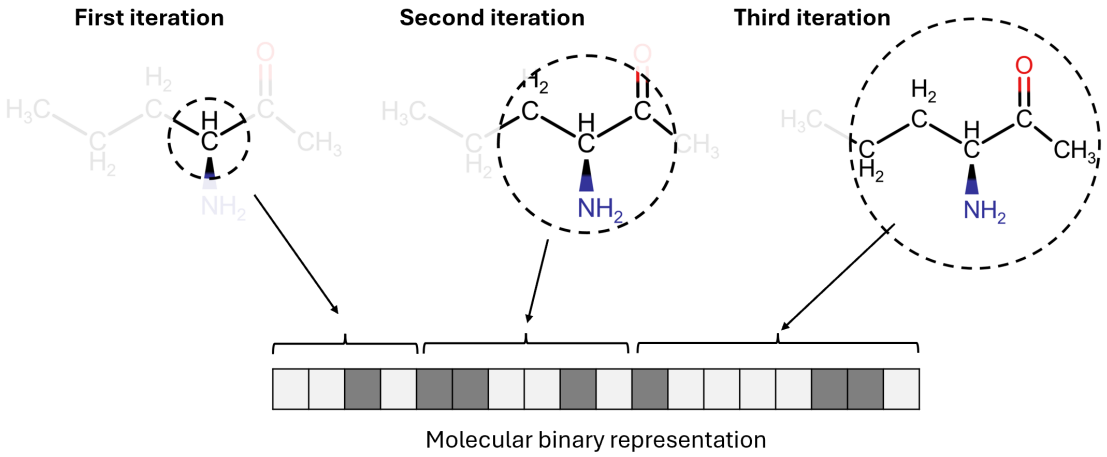


Figure 5: ECFP generation through iterative encoding of atomic environments with increasing radius, resulting in a fixed-length binary fingerprint.

of fragment-based representations, but one of the most commonly used implementations is available in RDKit. This approach extracts 85 predefined fragments from the *rdkit.Chem.Fragments* module, producing an 85-dimensional numerical array.

Fragprints

This vector representation is composed by the simple combination of ECFP and fragment representation form RDkit. By concatenating both vectors a new representation is generated expanding the content of information[Gri+23]. The motivation behind this approach lies in the corresponding strengths of each method: fragment descriptors tend to perform better in scenarios where functional groups play a critical role in prediction, while ECFP generally provides stronger overall performance. As a result, Fragprints performed better than ECFP and Fragment representation for the transition wavelength prediction task with GP [Gri+22].

$$\mathbf{x_{fp}} = \begin{pmatrix} 1 & 0 & \cdots & 1 \end{pmatrix}, \quad \mathbf{x_{fr}} = \begin{pmatrix} 8 & 1 & \cdots & 2 \end{pmatrix}, \quad \mathbf{x_{frp}} = \begin{pmatrix} 1 & 0 & \cdots & 1 & 8 & 1 & \cdots & 2 \end{pmatrix}$$

This opens up the possibility of designing new molecular vector representations, by simply concatenating existing ones that are best suited for our task.

Molecular descriptor

Accordingly to the definition given by [TC08]:

> A molecular descriptor is a result of a logical and mathematical transformation, in which the chemical information is put into a number or a result of a standardized experiment. This way the chemical data can be used in a qualitative or quantitative analysis.

Molecular descriptors might not be entirely considered as representation methods in the context of Machine Learning, since they are structured on the value of chemical experiments, instead of completely relying on the molecule structure [OHR22]. That being said, several molecular descriptors have been developed and are open source, but during this research the most encountered one, was the molecular descriptor called Mordred [Mor+18]. The Mordred descriptor describes molecules by physical, chemical and electronic properties in numerical values. Multiple researches state that Molecular descriptors outperform other representations (Vector, Graph, String, etc...) for virtual screening tasks [OHR22] [Tom+23], which is why they will be considered for later implementation.

3.2.2 String representation

SMILES

Stands for Simplified Molecular-Input Line-Entry System (SMILES) and represents molecules with a string build from ASCII characters, first invented in 1988 by David Weiniger to address the arising challenges to express molecules on computing systems [Wei88]. Atoms are expressed with the letter of the periodic table and bonds are indicated with the symbols: ['-': Single bond], ['=': Double], ['#': Triple], ['$': Quadruple]. Rings are represented by numbers, which accompany the starting and closing elements, with the same number randomly assigned to both. An example is the representation of benzene: [C1=C-C=C-C=C1]. Many other aspects, such as branching, nested branching and aromaticity can be represented as well in a SMILE string.

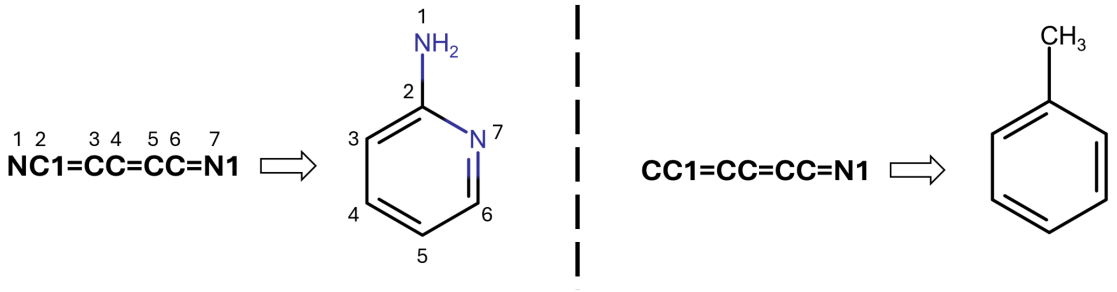


Figure 6: Conversion of SMILES strings into molecular graphs, illustrating how string notations are mapped to molecular structures.

Because of the relative easy human interpretation and writing, SMILES are one of the most popular string representations in cheminformatics. However, a problem arises from the fact that the same molecule can be written in different SMILES strings, which makes database searches and analysis more challenging, among other issues [WGL22].

SELFIES

One major disadvantage of SMILES representation is the large fraction of strings that do not represent any valid molecules, making the usage difficult for the discovery of novel materials. Self-referencing Embedded String (SELFIES) were introduced to resolve that issue, where each string representation corresponds to a valid molecule [WGL22], showing its value to explore large chemical spaces due to the combinatorial explosion of possible structures. The generation of SELFIES is defined by multiple derivation rules that prevent the possibility of invalid syntactical strings, making them immune to random mutations that could lead to invalid structures [Kre+20]. SELFIES was primarily developed to enable the generation of novel molecules using Variational Auto Encoders (VAE) and Generative Adversarial Networks (GAN). The diversity generated through GAN using SELFIES was 4 times bigger than with SMILES [Kre+20]. The key aspect is that SMILES can be made consistent by deriving them to SELFIES and, after enhancing alterations, transform them back to SMILES.

SELFIES was primarily developed to enable the generation of novel molecules using VAE and GAN

3.2.3 Graph representation

A convenient and intuitive way of representing molecules is through undirected graphs $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where each node corresponds to an atom $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$ and the edges indicate the type of bond between the atoms $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. These graphs are called "attributed molecular graphs" and have become an essential molecular representation for many AI-based models [WGL22], [Xu+10]. An alternative is to express molecular structures in a reduced graph, where nodes represent substructures, allowing to concentrate on the crucial compositions [Frö+06]. By default, without performing any computations, graph representations can be directly used in Graph Neural Network (GNN) or convolutional GNN. However, many studies implement molecular graph matching by employing kernel functions designed specifically for graphs in regression and classification tasks [Xu+10], [Frö+06].

When it comes to types of molecular representation, the variety is huge, providing many alternatives for the customization of a ML model. As for this master

thesis, the idea is to benchmark most of them to see how they perform compared to each other.

3.3 Machine learning methods

Data representation is one of the key components in developing data-driven ML models. However, once molecular data has been appropriately encoded, the next important decision is the selection of a model suitable for the task at hand. In cheminformatics, a diverse set of models have been employed successfully, each with specific strengths and weaknesses. This flexibility, while valuable, poses a challenge with many options, which is: how do we determine the best model for a specific, constrained scenario?

For general-purpose prediction tasks, classical ML models like Support Vector Machines (SVM)s and Random Forest (RF) have demonstrated solid performance in both regression and classification tasks across various molecular datasets [Mah+20; Tom+23; Zha+06]. Meanwhile, deep learning models, including GNNs, Recurrent Neural Networks (RNN)s and others, have gained popularity due to their scalability and capacity to learn complex representations, especially when trained on large datasets [WB20; Elt+19; Ral+05].

Recent advances have also introduced the use of transformer-based large language models (LLMs), such as ChatGPT, demonstrating promising performance even in domains like predictive chemistry, despite not being originally designed for such applications [Jab+24]. In parallel, unsupervised and reinforcement learning methods have been increasingly utilized in chemical space exploration and molecular design tasks, providing novel strategies that extend beyond traditional supervised learning approaches [PIT18].

While predictive performance remains important, the need for uncertainty quantification, as in this work, makes probabilistic models essential. These include frequentist approaches like RF with bootstrapping or deep learning models using Monte Carlo-dropout, and Bayesian methods, such as Bayesian neural networks and GP, which offer principled uncertainty estimates [Gri22; Sno+15; Tom+23].

Despite the diversity in modeling strategies, the selection of a suitable model cannot rely entirely on algorithmic capability. Practical considerations such as dataset size, computational resources, training time, and the nature of the prediction task must also guide the choice. This is particularly relevant in real-world molecular property prediction tasks, where data is often scarce due to the high cost of experimental labeling. Unlike benchmark studies that assume access to large, curated datasets, this research aligns with more constrained settings.

A comprehensive benchmarking study on low-data chemical datasets [Tom+23] evaluated five probabilistic models: RF (using NGBoost), GP, spectral normalized

GP (SNGP), GNNs, and Bayesian neural networks, across three representations (fingerprints, graphs, molecular descriptors) and multiple datasets. For regression tasks (BioHL, $n = 150$; Freesolv, $n = 637$; Deaney, $n = 1116$), GP and RF models using molecular descriptors outperformed others, particularly in extremely low-data regimes. In classification tasks (BACE, RBioDeg, BBP; $n \approx 1500$–1800), all models performed comparably, though GP and RF again slightly outperformed their counterparts.

These findings contrast with results from large-data studies like those on the QM9 dataset ($n \approx 134$k molecules), where more complex, parameter-rich models such as deep neural networks performed best [Fab+17; GSC21]. This reinforces a key insight: model complexity must align with data availability. Models with high capacity require substantial data to generalize effectively, a luxury often absent in practical cheminformatics applications.

With this context in mind, this thesis employs BO as a strategy to guide molecular selection and evaluation. BO depends highly on a probabilistic surrogate model to estimate both predictions and uncertainty, enabling efficient prioritization of candidates based on expected improvement or information gain [GSC21].

While Bayesian Neural Networks have often been used as BO surrogates, their performance degrades significantly in low-data scenarios. In contrast, GP offer a more attractive alternative. GP not only delivers calibrated uncertainty estimates but also provide a flexible configuration space through their choice of kernels and hyperparameters, making them particularly well-suited for surrogate modeling under limited data [Gri22; BW22; Wu+19]. Though RF have also shown promise as BO surrogates, GPs offer a greater modeling power and flexibility, solidifying their role as the optimal choice for this study.

3.4 Libraries

Many researches have been made when it comes to the application of Machine Learning in the field of chemistry and biology, all of them have contributed to the development and creation of various codes to enable the application of the methods mentioned above. Moreover, experts have come together to make collaborations to design frameworks that facilitate the use of intelligent algorithms and the search for new molecules, so called libraries. These libraries provide the user with the necessary tools to implement the desired ML-models without the need to develop everything from the start. This section provides a brief overview of several toolkits that have significantly helped advance the use of artificial intelligence in computational chemistry.

Cheminformatics Libraries

RDKit is probably one of the most widely used libraries in the world of cheminformatics. It contains a large number of functions that allow the manipulation of molecules as well as their representation for its use in different intelligence models. RDKit is designed to integrate easily with other libraries that provide tools for building and applying machine learning models. Apart from that, it offers users an easy access to large datasets making it indispensable for all kind of AI-Workflows. [RDK25]

DeepChem is one of the most complete packages for the scientific use of artificial intelligence with a focus on drug discovery, but also serves as a great tool for applications in materials science, bioinformatics and many more. DeepChem comes with numerous algorithms for tasks like molecule classification, property prediction, generative modeling, among many others. [Dee25]

Gauche is a library explicitly designed and developed for the exploration of chemical spaces using probabilistic machine learning methods, offering a practical tool for materials scientists and chemists engaged in virtual screening. The library is optimized for the use of Gaussian processes and the prediction of density functional theory (DFT) calculations. In addition, Gauche interfaces with libraries such as GPyTorch and BoTorch to provide an accessible and flexible platform to develop new algorithms for probabilistic modeling and optimization in chemistry.[Gri+23]

GP & BO Libraries

GPyTorch is one of the most used python libraries for statistical modeling introducing, a framework for the creation of GP models based on top of PyTorch. Traditionally, implementing GP models required manually defining and customizing core mathematical routines, such as the marginal log likelihood for training. GPyTorch significantly simplifies this process, enabling rapid prototyping and experimentation with GP models. Another key advantage lies in its ability to utilize modern hardware to speed the inference and prediction process through the use of Blackbox Matrix-Matrix (BBMM), a method that reduces computational complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$. Additionally, the library includes multiple predefined GP components like kernels, model types, likelihoods and many more.[GPy25]

BoTorch similar to GPyTorch, it is built on top of PyTorch and it is a library dedicated for BO, providing a versatile framework for the coding of probabilistic models, acquisition functions, and optimizers. Given that BO heavily relies on surrogate models such as GP, BoTorch is fully compatible with GPyTorch, enabling seamless

integration of GP models for solving complex optimization problems. Important customization features related to BO such as acquisition functions are predefined in a way that they are easy to adjust and scale for the task at hand. This flexibility and modularity allows researchers to focus on the development and design of such models, rather than on the complexities of computational implementation.

3.5 Related work

Before we deep deep into the development of a BO Framework for the acceleration of material discovery, specifically applied to the scenario of finding the photoswitch with the highest transition wavelength among a pool of candidates, a short review of similar tasks will be presented on existing approaches.

There are several studies on the application of machine learning for accelerating material discovery and virtual screening, the focus here will be on research that utilizes BO with GP as surrogate models, specifically in the context of small datasets.

One notable study focused on a screening scenario, evaluating 720 ingredients (additives) across different reactions, with the goal of identifying the reaction that maximizes the UV210 area absorption. The strategy involved evaluating a variety of GP and BO combinations to establish an optimal screening model. Key components considered in the study included different kernels, acquisition functions, and data representations. Results showed that best performance where obtained using a Matern kernel, UCB as acquisition function, and reaction fingerprints (DRFP), which are binary high-dimensional representations. [Ran+24]

Another related study performed was the search of stable crystal structures among a dataset. Using BO and GP as surrogate model, they would try to find the molecule with the lowest total energy (a low total energy implies more stable) in a 150 dimensional chemical space. Outcomes were that besides searching in a high dimensional space, the BO model was able to outperform by up to 39% in comparison to a random search. [Yam+18]

A more concrete approach to optimizing material discovery with BO involved implementing a novel acquisition strategy that dynamically balances exploration and exploitation. Initially, the method emphasizes exploration to gain a broad understanding of the search space, then shifts to exploitation once uncertainty is reduced. This adaptive strategy demonstrated clear improvements over single-method approaches, especially when starting with limited initial data [Rai+24].

BO is a very effective logarithm in scenarios where the evaluation of samples is highly costly, contributing to small dataset, and where the correlation between candidates (in this case molecules) is complex and unknown [JK23]. In summary, BO combined with GP provides an efficient and powerful approach for navigating complex, costly, and data-limited environments, making it the preferred choice for

tasks like molecular optimization.

4 Methodology

This chapter explains the methods used to speed up the discovery and optimization of azobenzene-based photoswitches. It describes how Bayesian optimization and Gaussian processes were applied to analyze data and predict photoswitch properties. The chapter also covers how different molecular representations, kernels, and settings will be tested and adjusted to improve the accuracy and efficiency of the process.

4.1 Model structure

This section presents a detailed breakdown of the structural development for the model, defining the key framework components and how they interact with each other. The core concept of this model for accelerating the molecular discovery is the combination of GP with BO, functioning as a sequential learning environment.

Data is split in two sets: the training set (data that has been observed), and the holdout set (candidate pool). As the model operates in a sequential learning cycle, the process is repeated in a loop, where each repetition corresponds to an iteration. The first step in the iteration involves the similarity computation between training inputs using a kernel (covariance function). Following this, the defined loss function ,typically the marginal likelihood, and an optimizer are employed to determine the most adequate hyperparameter values for the kernel function. Once the surrogate model has been fitted to the training data, mean and uncertainty predictions are computed for all the test samples, providing a statistical model for all considered candidates. In the context of the presented application scenario, each photoswitch derivative in the candidate pool receives a wavelength prediction with an associated uncertainty value. Based on the information provided by the GP, the chosen acquisition function within the BO framework computes the value of each candidate in terms of its potential to identify the optimal molecule. The candidate with the highest acquisition value is selected for evaluation, removed from the selection pool, and added to the set of observed data. The process then starts again from the beginning, now incorporating the new observation in the training set.

After each iteration we expect to get a better image of the objective function within the chemical searching space, helping the BO to make better decisions every time. In the **Figure 7** the iteration workflow of the framework is graphically illustrated. As a side note, the statistical function computed by the GP and the acquisition function from BO are discrete and cannot be accurately represented in a continuous diagram, such as the one shown. The illustration serves only to aid in the interpretation of the process.

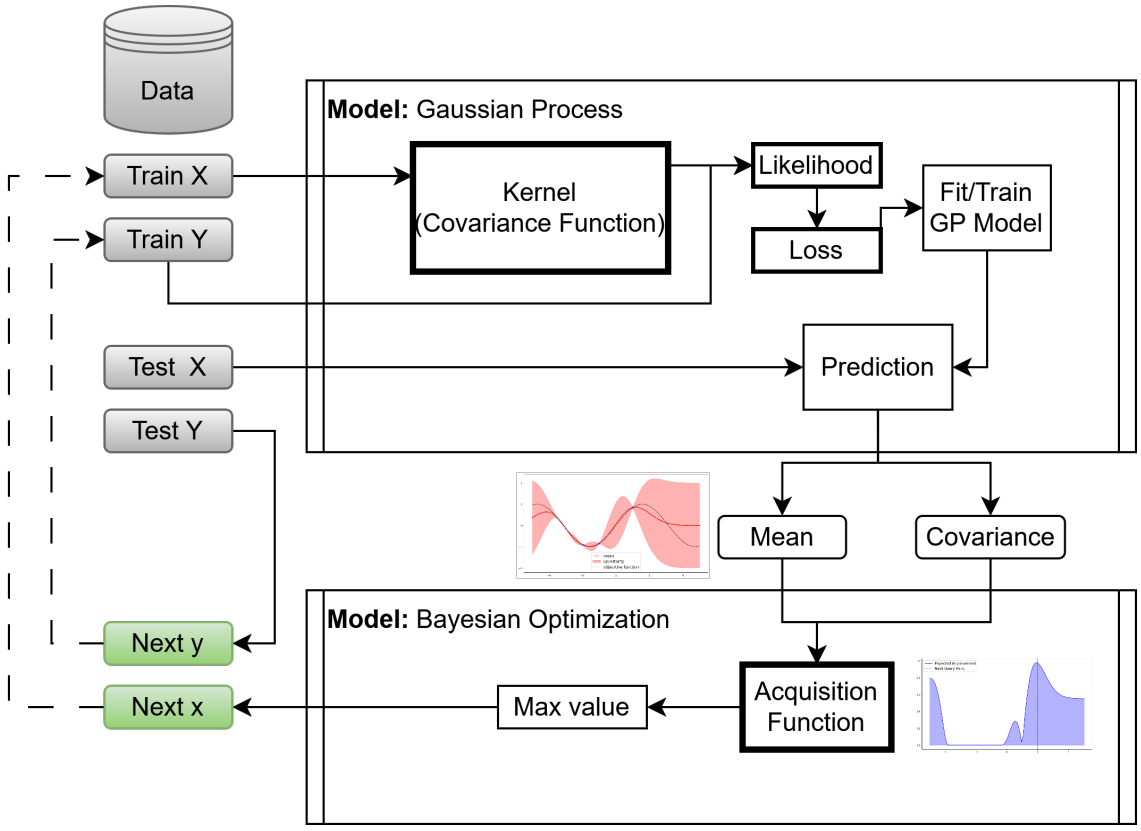


Figure 7: General structure of active search for optimal photoswitch molecules, combining a GP surrogate model with a BO process. The training set includes evaluated molecules, the test set contains candidate molecules, and the selected molecule (next x) is evaluated to obtain (next y) in each iteration, which are then added to the training set for the next iteration.

In the workflow diagram, several components are highlighted with a thicker line, underscoring the key factors that will be considered for optimizing the algorithm such as the choice of kernel function, loss definition, and acquisition function. Other important aspects, though not highlighted, include the selection of a molecular representation and the initialization data.

4.2 Data collection and Preparation

For the initial experiments, the data pool will consist of the photoswitch dataset, which contains a total of 405 molecules labeled with their respective transition wavelengths. This dataset was carefully curated by Ryan-Rhys Griffiths for the study *"Data-driven discovery of molecular photoswitches with multioutput Gaussian pro-*

cesses" [Gri+22].

Two factors outside the learning model significantly influence the overall framework: the data representation and the size of the starting set. The first factor is the molecular representation of photoswitches, illustrated in **Figure 8**. The representations analyzed will include vector-based methods (Fingerprints, Fragments, and Fragprints), string-based methods (SMILES), and graph-based methods. It is important to note that for each type of molecular representation, different kernels are more appropriate for computing similarity between inputs. However, despite the differences in representation, the values produced in the Gram matrix by the kernel function are always numerical, meaning that the subsequent modeling process remains identical across representations. Thus, the only factor that must be adapted based on the molecular representation is the choice of kernel.

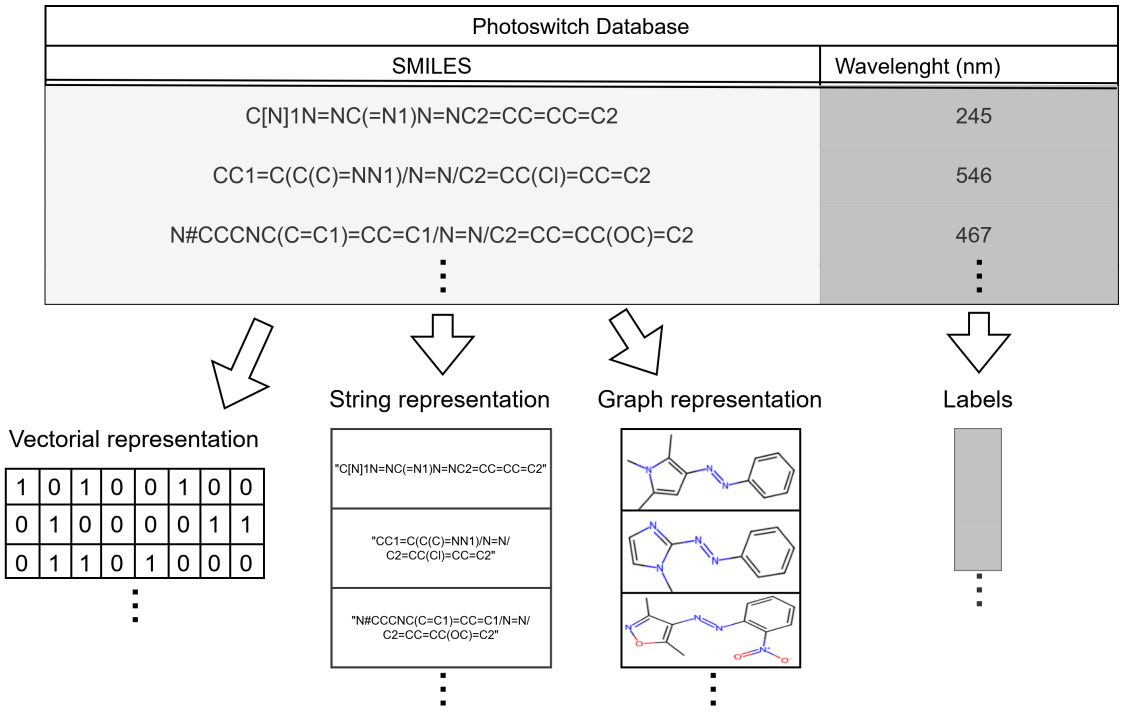


Figure 8: Molecules in the photoswitch dataset are featurized as SMILES. Within the BO framework, molecular representations can take the form of high-dimensional vectors, strings, or graphs. Each representation type requires a specifically designed kernel to handle its unique structure.

Another crucial aspect is the selection of the initial set of observed samples (known molecules used to construct the surrogate model) and the set of holdout samples (the pool of candidate molecules). The difference of starting with 10 observed samples

versus 1 sample has a significant impact on the performance of BO in searching for the optimal molecule. In the domain of material discovery, it is common for experts to initiate the screening process without prior information, selecting a molecule at random for evaluation and allowing the framework to proceed from there. However, the use of BO with such limited initial data can be counterproductive, as the GP may struggle to construct a reliable statistical model. This, in turn, can mislead the optimization process and significantly reduce its effectiveness [Ran+24].

Since the main objective is to identify the optimal molecule with the fewest possible evaluations, different selection strategies will be explored when starting with a predefined set of molecules. Rather than selecting a fixed number of samples at random, approaches such as clustering-based initialization and maximal coverage initialization will be applied to help the GP build a more effective surrogate model and better represent the molecular latent space.

4.3 Bayesian optimization Framework

Now that the main framework of the BO has been defined and explained, a deeper examination on the individual components will be made. Specifically, the goal of this section is to identify the most promising methods for each part of the ML model. This begins with an analysis of the effectiveness of the statistical models created by the GP and their calibration, followed by the selection of suitable kernels for different representations, as well as appropriate acquisition functions.

4.3.1 Gaussian process as Surrogate Models

As mentioned in previous chapters, GP are particularly effective when labeled data is scarce and little is known about the objective function. However, GP also face significant challenges when applied to high-dimensional spaces [BW22; Gar23]. The underlying reason is intuitive: consider a chemical space with three dimensions, each bounded between -1 and 1. To have a measurement at each corner of this space, 2^3 observations would be required. If instead the chemical space consists of 100 dimensions, the number of required observations escalates drastically to 2^{100}, illustrating the well-known *curse of dimensionality*. Additionally, increasing the dimensionality has a direct impact on the behavior of distances between points, an aspect that will also be discussed in the context of kernel functions [BW22].

For both string and graph representations, the dimensionality depends primarily on the kernel used for the GP rather than on the representation itself. Of course, larger strings and graphs lead to a higher-dimensional space, but the relationship is not as straightforward as it may seem. The risk with the use of kernels lies in the kernel-induced feature space, where if the kernel function is too simple, it may fail to distinguish between different molecules [XSA05]. Another option is to use methods

like Variational Autoencoders (VAE) and other ML techniques to map molecular strings and graphs into an optimal latent space, which can later be passed to a kernel. However, this approach requires a significant amount of data to learn an appropriate representation, which is not available in this scenario [Tom+23], [Kre+20].

For vector representations the curse of dimensionality is clear. ECFP are binary vector representations of the length 2048, making them part of a very high-dimensional chemical space at first sight. Yet, high-dimensionality takes another meaning with binary representations, as it may translate to lower effective dimensionality compared to other vectors using continuous representations in a Euclidean space [Ran+24].

4.3.2 Kernel selection

The kernel selection is also among the key topics of this research; however, there is not a methodological process to define the perfect kernel for multiple reasons. As mentioned before, for each molecular representation used, a different kernel has to be applied to compute the similarity. Therefore, the kernel selection will be handled individually for each representation. The second aspect is the lack of expressiveness or accessibility of the objective function, which behaves as a black box. As a result, there is no straightforward mathematical or analytical way to determine which one fits best for the task. For this reason, a list of kernels will be applied and evaluated, primarily through trial and error. The main focus will be on kernels for vectorial representations, as they offer a greater variety and are generally the most versatile for GP.

Kernels for vector representation: In this part two types of kernels have to be analyzed: Similarity kernels for continuous representations and similarity coefficients for binary vectors.

For continuous representations, the kernels that will be used are:

- **Radial Basis Function (RBF):** Known as one of the most used Kernels for GP [WR06].
- **Matérn Kernel:** A kernel that provides a solution for the *"to much smoothness assumptions"* provided by the RBF [Ste99].
- **Dot Product Kernel:** This kernel is a simple and commonly used kernel in linear regression models for molecular descriptor comparisons.

All of these kernels, along with their corresponding formulas, were previously introduced in the background section **2.1.2**.

For binary representations, similarity kernels are different. The task for measuring similarity between binary representations might have started in 1901 by Jaccard [Jac01]. Since then, multiple binary distance measures have been designed for different applications. Various studies have been made to test the efficiency of similarity kernels for ECFP representations [CCT+10]. Based on various researches [Tod+12], [Swa+05], [Wij+16] a list of kernels will be tested to optimize the performance of the BO for the photoswitch set. These similarity kernels are primarily designed for binary representations but can also be applied to integer or real-valued vectors, such as the fragprint representation, which combines both binary and integer values.

For binary-integer representations, the kernels that will be used are:

- **Tanimoto Kernel:** This kernel, also known as the Jaccard similarity, is widely used for comparing binary fingerprints in cheminformatics [Swa+05]. It is defined as:
$$k_{\text{Tanimoto}}(x, x') = \frac{\langle x, x' \rangle}{\|x\|^2 + \|x'\|^2 - \langle x, x' \rangle}$$
- **Dice Kernel:** Similar to the Tanimoto kernel but gives more weight to shared features [CCT+10]. It is defined as:
$$k_{\text{Dice}}(x, x') = \frac{2\langle x, x' \rangle}{\|x\| + \|x'\|}$$
- **Frobes Kernel:** This kernel is derived from the Frobenius inner product, often used in matrix comparisons but applicable to binary vectors as [Tod+12]:
$$k_{\text{Forbes}}(x, x') = \frac{n\langle x, x' \rangle}{\|x\| + \|x'\|}$$
where $x \cdot x'$ is the standard inner product.
- **Inner Product Kernel:** A simple linear kernel that measures similarity based on the dot product:
$$k_{\text{Inner}}(x, x') = \langle x, x' \rangle$$
This kernel assumes a high similarity when binary vectors have many common active bits [Swa+05].
- **MinMax Kernel:** This kernel is based on the minimum and maximum of corresponding elements in binary vectors:
$$k(x, x') = \frac{\sum_{i=1}^{d} \min(x_i, x'_i)}{\sum_{i=1}^{d} \max(x_i, x'_i)}$$
It is useful for comparing feature sets where presence or absence matters [Swa+05].

- **Sorgenfrei Kernel:** This kernel is based on the Sorgenfrei similarity, defined as:

$$k_{\text{Sorgenfrei}}(x, x') = \frac{\langle x, x' \rangle^2}{\|x\| + \|x'\|}$$

  It provides a normalization similar to Tanimoto but focuses on the larger of the two vectors' magnitudes [CCT+10].

Kernels for String Representation: When it comes to creating kernels for string similarity, the possibilities are virtually limitless. These kernels are commonly used in text representation models to capture contextual relationships. One widely used and intuitive kernel is the *bag of n-grams*, where text similarity is computed by matching multiple contiguous sequences (such as words, elements, or letters) found in the two texts being compared [Li+17]. The similarity between the texts is then evaluated based on the number of sequences they share.
Note to the reader: Although the following discussion addresses aspects of string kernels, practical limitations in memory usage and computational complexity ultimately prevented their full implementation. Nevertheless, since their study was part of the research, they are presented here for completeness.
Let's compare the words **"hello"** and **"hero"** using **bigrams** (n=2). Meaning that a list of continuous subsequences of length n=2 will be derived from each word.

"hello" → {"he", "el", "ll", "lo"}
"hero" → {"he", "er", "ro"}
Common bigram: "he"

Since they share one bigram, their similarity is computed based on the proportion of shared sequences. If we instead use **trigrams** (n=3), we get:

"hello" → {"hel", "ell", "llo"}
"hero" → {"her", "ero"}

Here, there are no shared trigrams, leading to a lower similarity score.
This example illustrates how the *bag of n-grams* method captures similarity by identifying overlapping sequences in text. Of course, this is a very simple example and, as mentioned before, the configurations that can be made in string kernels are endless. For instance, instead of just considering sequences of length n, the kernel could take into account all sequences from length 1 to n. Another option would be to include a weighting factor that assigns more importance to longer sequences compared to shorter ones, and so on.

Since the application of string kernels will be applied on molecular string representations like SMILES, the kernel used will be a Subsequence String Kernel (SSK), as proposed by Lodhi et al. [Lod+02]. This kernel was originally motivated by text classification tasks and has been successfully tested in Bayesian optimization (BO) on SMILES [Mos+20], obtaining great results in comparison to other machine learning models. The string kernel $k_n(a, b)$ between the strings a and b of n-order and is expressed as:

$$k_n(a, b) = \sum_{u \in \Sigma^n} c_u(a) c_u(b),$$

where

$$c_u(s) = \lambda_m^{|u|} \sum_{1 < i_1 < \cdots < i_{|u|} < |s|} \lambda_g^{i_{|u|} - i_1} \mathbb{1}_u((s_{i_1}, \ldots, s_{i_{|u|}})).$$

Σ^n is the list of sub-sequences and $c_u(s)$ measures the value of the sub-sequence u within the string s. The hyperparameters λ_m and $\lambda_g$represent the match decay and gap decay factors. These parameters account for the continuity and discontinuity of character sequences in SMILES representations, enabling the kernel to robustly compute structural similarities between molecules.

Example of the SSK with two SMILE strings:

$$\text{For } a = [\text{CCO}]\text{: subsequences } u \text{ are: } \{\text{CC}, \text{CO}\}$$
$$\text{For } b = [\text{CCN}]\text{: subsequences } u \text{ are: } \{\text{CC}, \text{CN}\}$$

For $n = 2$:

$$k_2(a, b) = c_{\text{CC}}(a) \cdot c_{\text{CC}}(b) + c_{\text{CO}}(a) \cdot c_{\text{CN}}(b)$$

The issue with this kernel is the computational cost that scales $\mathcal{O}(nl^3)$. However an approximation kernel designed by Henry B. Moss [Mos+20] that splits sequences in m parts reduces the complexity by $\mathcal{O}(nl^3/m^2)$ making use of parallel calculations, which is integrated in the GAUCHE library for its application.

Kernels for graph representation: Finally, the task of computing similarity between molecules involves generating a Gram matrix over a set of graph representations.

$$k(G_i, G_j) = \langle \phi(G_i), \phi(G_j) \rangle_{\mathcal{H}}$$

This type of kernels can be interpreted as an inner product kernel with molecular graphs G that has been mapped through $\phi()$ into a embedding feature space, like it

is the case of ECFP [Ral+05]. The challenge resides on the direct 'raw' similarity measure making use of the information provided by their nodes (elements) and edges (bonds). Graph kernels can be divided into main functions[GFW03] :

- **Kernels based on labeled pairs** focusing only on the labels of the start and end nodes of walks, not considering the sequence in between.
- **Kernels Based on Contiguous Label Sequences** capturing the entire sequence of the walks, instead of just considering the start and end.

The second type of kernel is capable of capturing more complex relationships and detailed structural information within the graph compared to the first. This is because it takes into account the entire sequence of nodes in the walk, rather than just the start and end points. However, this increased expressiveness comes at the cost of higher computational complexity. One of the most widely used kernels in chemoinformatics is the Weisfeiler-Lehman Optimal Assignment kernel. This kernel is motivated by the Weisfeiler-Lehman test for graph isomorphism, combined with the use of a strong base kernel that guarantees the creation of a positive semi-definite Gram matrix K [KJM20].

Consider to graphs G and H with the set of nodes $\mathcal{N}(G) = \{u_1, u_2, \ldots, u_n\}$ and $\mathcal{N}(H) = \{v_1, v_2, \ldots, v_m\}$, where $\mathfrak{B}(G, H)$ is the set of all possible bijections between the two graphs. If both graphs contain the same number of nodes $n = m$, the set size of bijections is $n!$. In case that one graph contains more nodes $n > m$ than the other, the bijection set size is of $\frac{m!}{(m-n)!}$ where nodes from the bigger graph are left without a pair.

Additionally, on each graph a Weisfeiler-Lehman label refinement is performed for h iterations, leading to a sequence of labels for each node: $\tau_0(u), \tau_1(u), \ldots, \tau_h(u)$.

Using this refined labeling process, the Weisfeiler-Lehman kernel is defined as:

$$K_{\mathrm{WL}}(G, H) = \max_{\pi \in \mathfrak{B}(G,H)} \sum_{i=1}^{\min(n,m)} k(u_i, v_i)$$

where $k(u_i, v_i)$ is the base kernel between two nodes, counting with a Dirac function δ the number of matches across different refinement levels, and π denotes the bijection that maximizes the sum of the base kernel values.

$$k(u, v) = \sum_{i=0}^{h} \delta(\tau_i(u), \tau_i(v))$$

4.3.3 Acquisition function design

After configuring the GP to serve as an effective statistical model, the next crucial step is selecting an appropriate acquisition function. It is important to note that, because the optimization is performed over a finite set of candidate molecules rather than a continuous space, both the statistical model provided by GP and the acquisition function from BO operate in a discrete setting. As a result, gradient-based optimization methods cannot be used to find the optimum of the acquisition function. Instead, each candidate is scored individually, and the one with the highest acquisition value is selected for evaluation.

The primary acquisition functions considered for the BO framework are: The UCB function, which is characterized by its simplicity and exploitative nature, and can be adjusted by the λ hyperparameter. And the EI, which inherently possess a more exploitative behavior. However, the exploration-exploitation balance can be adapted by the hyperparameter ξ [Gar23].

In addition to these two acquisition functions, a few others will be tested as well to improve performance. Because the objective function behaves as a black box, offering no explicit form or gradient information, the most practical way to determine the most effective acquisition function is through trial and error. With the pre-defined functions provided by the BoTorch library [BoT25], it will be uncomplicated to evaluate these functions across multiple scenarios.

However, similar to the paper *"Accelerating material discovery with a threshold-driven hybrid acquisition policy"* [Rai+24], the idea is to combine multiple acquisition functions for different iteration numbers as well as different initialization scenarios. The main goal of this research is to identify the optimal molecule with the fewest measurements possible, starting with initial observations (the initial scenario) and following with a series of sequential evaluations. The study by Tom, Gary *et al.* [Tom+23] suggests that starting with a smaller initial set of observations can improve the efficiency of BO. However, having too few data points early on may lead to a less accurate surrogate model, which can negatively impact the quality of initial decisions. Part of this research will focus on finding the balance between initialization sets and sequential selection. For this reason, initialization algorithms that methodically choose starting samples will also be considered. The idea of these selection algorithms is to achieve the best possible coverage of the chemical space with a limited number of observations, using techniques such as clustering and maximal dissimilarity methods.

4.4 Model tuning and evaluation

The performance of the BO framework, will highly depend on all the presented components when searching for the optimal molecule. Therefore, this section summarizes all possible combinations, such as the choice of representation, kernel, and

other factors for model tuning.

Representation	**Kernels**
Vector: ECFP, Fragments, Fragprints, Morderd	RBF, Matern, Linear, Tanimoto, Dice, Forbes, Inner Product, MinMax, Sorgenfrei
String: SMILES, SELFIES	Sub-sequence string
Graph: Edgelist	Weissfeiler-Lehman

Table 1: Molecular representation types and their corresponding kernels considered in the BO framework. Direct use of string representations with the SSK could not be fully implemented.

Molecular representation types and their corresponding kernels considered in the BO framework. Direct use of string representations with the SSK could not be fully implemented.

As mentioned earlier, the choice of molecular representation is closely tied to the selection of the kernel. **Table 1** summarizes the list of molecular representations alongside the corresponding kernels that can be applied. At first glance, it is evident that there is a primary focus on comparing vector representations with various kernels. This focus stems from the fact that molecular vectors are among the most commonly used representations for virtual screening, making them especially attractive for implementation in future tasks and datasets. Furthermore, the implementation of string and graph kernels is more complex, which is why only one kernel for each of these representations will be considered in this research.

Moving on to the choice of acquisition functions, initial set selection, and the number of initial observations, these components are intended to optimize the selection process, as shown in **Table 2**. These factors are independent of the chosen kernel and representation, and therefore will be addressed separately.

Each scenario with a different number of starting observables will behave differently in the selection process. While the surrogate model will likely have no trouble inferring the objective function in the 10% and 5% scenarios, when starting with only 1% or a single sample, the GP will struggle to provide enough guidance for the BO during the early iterations. This will lead to slower convergence and potentially less reliable predictions.

The surrogate model provided by the GP is likely to be the most ambiguous com-

Acquisition function	**Prior selection process**	**Training number**
UCB, PI, EI	Random, Cluster, Max distance	1 Sample, 1%, 5%, 10%

Table 2: Overview of the main acquisition functions, prior selection algorithms, and training set sizes that will be tested for active search initialization.

ponent, making its behavior difficult to interpret. Therefore, it will be essential to define specific GP prediction metrics to assess how accurately each statistical model approximates the objective function at each iteration

One of the most used and simple methods to evaluate regression models is the mean squared error loss [WR06], measuring the error average between predictions and actual values. Another alternative is the mean absolute error (MAE), which is less sensitive to large outliers. While both metrics are commonly used in various studies [Gri+23], [Ran+24], they are sensitive to the scale of the target variable and can be distorted by differences in magnitudes. As a result, the coefficient of determination R^2 presents itself as a better metric for prediction and performance monitoring. The $\mathbf{R^2}$ score is defined as:

$$R^2 = 1 - \frac{\sum_i (y_i - \mu(x_i))^2}{\sum_i (y_i - \bar{y})^2}$$

where y_i is the actual value, $\mu(x_i)$ is the predicted value and $\bar{y}$ is the mean of the actual values. The metric R^2 ranges from 1 to $-\infty$, where 1 indicates perfect predictions, 0 means that the model is no better than predicting the mean and values below 0 perform even worse.

There is one disadvantage for this metric: it is not taking into consideration the uncertainty that GP are able to provide in comparison to other regression models. Therefore a metric that is used for GP is the: Mean Standardized Log Loss (MSLL) composed by the log probability of the GP regression at each test point x^* and a trivial model formed by the train observations y^* [WR06]:

$$-\log p(y^*|D, x^*) = \frac{1}{2}\log(2\pi\sigma^{*2}) + \frac{(y^* - \bar{f}(x^*))^2}{2\sigma^{*2}}$$

$$\mathrm{SLL}(x^*) = \log p(y^*|D, x^*) - \log p_{\text{trivial}}(y^*|D)$$

where y^* is the true observed value at the test point x^*, $\bar{f}(x^*)$ is the predicted mean at x^*, and σ^{*2} is the predicted variance (uncertainty) at x^*.

The MSLL is computed by subtracting the trivial model from the trained model, which corresponds to the standardized log loss (SLL), and then averaging over all N test points [GPy25].

$$\text{MSLL} = \frac{1}{N}\sum_{i=1}^{N}\text{SLL}(x_i)$$

For MSLL values over 0 the GP model performs worse than the trivial model. If the value is negative, it implies that the model performs better than the trivial model.

Although MSLL can be a strong indicator of GP performance, especially in terms of uncertainty calibration, it may not be as meaningful in the upcoming scenario. This is because the trivial model, which is based on the training data, may vary significantly from one trial to another, reducing its reliability as a consistent metric. Therefore when computing the MSLL for performance indicators, the trivial model is not to be considered, resulting into the following metric:

$$\text{MSLL} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}\log\left(2\pi\sigma_i^{*2}\right) + \frac{(y_i^* - \bar{f}(x_i^*))^2}{2\sigma_i^{*2}}\right]$$

Apart from the GP, the main goal is to optimize the number of iterations required to identify the best possible candidate from the pool. The fewer evaluations the framework needs to perform, the better. To achieve this, the model uses the variance and mean values generated by the acquisition function. These values provide important information on how the model balances exploration and exploitation, guiding its decision-making process in each iteration.

5 Experiments and Results

This section introduces the experimental procedure used to evaluate and test all the methods presented in the previous chapter to improve the selection process. The idea is to go step by step, where each test provides new insights that help guide the next actions, whether to discard some methods, apply certain combinations, or implement new algorithms that might enhance the process. To set the stage for this process, we begin with a brief outline of the main scenario components:

Problem Definition: The application of BO to sequentially identify the best molecules in order to obtain the optimal candidates with the highest transition wavelength from a given dataset.

Dataset: Consists of 405 photoswitch molecules represented using SMILES notation, each accompanied by an experimentally measured transition wavelength (in nanometers). It is divided into two parts: a training set, which includes the initially observed molecules, and a test holdout set, which contains candidate molecules that have not yet been evaluated.

Framework structure: Using the starting training set of observations, the surrogate model GP generates a first statistical model, which is passed to the BO, where for each candidate of the holdout set an acquisition value is computed. The molecule that has the highest acquisition value is evaluated and included to the training set as a new observable. This process is repeated until a certain amount of iterations is reached or a requirement has been achieved.

Building on this framework, the procedure of improvement will first reside on evaluating the best kernels in combination with their molecular representations, considering different training set sizes. This will first help to get a general overview on how well the surrogate model is capable to infer the objective function, thereby allowing the elimination of the least effective kernel-representation combinations.

Next step resides on the analysis of different acquisition functions generated from the statistical model, taking into consideration different training sizes using the same kernel and representation. The idea is to obtain information on how different policies behave for each surrogate model and to experiment with its hyperparameters in order to achieve the optimal balance between exploration and exploitation.

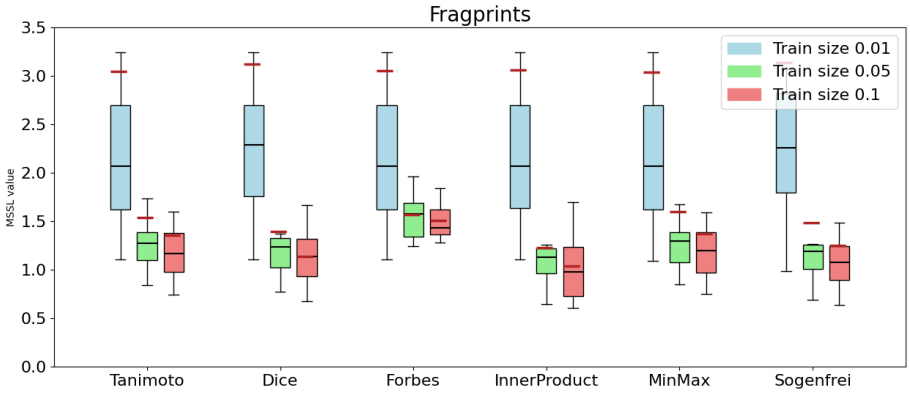
Once a general understanding is achieved and the best combinations for the BO framework have been identified, performance will primarily be evaluated based on the number of samples needed to find the optimal molecule.

5.1 Kernel and molecular evaluation

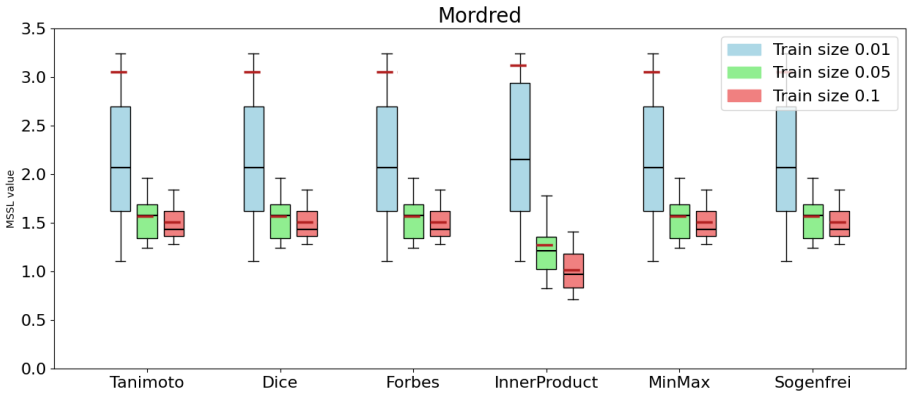
Different kernels and representations will be evaluated for generating the statistical model. By using the MSLL metric, valuable insights will be gained into how the GP performs with each combination. Since this process is independent of the other components, it presents a promising opportunity to obtain preliminary results without needing to implement the entire framework. This approach also allows the possibility of discarding some representations and kernels if necessary. Each of the combinations will be evaluated on tree different training set sizes: 1%, 5% and 10%.

To get an initial overview, box plots are used to understand the general behavior of different kernels with varying training sizes. The training points used to train the GP are randomly picked, and depending on which ones are chosen, the predictions can vary substantially. Therefore, the box plot diagram helps capture the average performance over multiple trials, minimizing the influence of random selection. Multiple things can be read from the graph **Figure 9**: Each box describes the model loss repartition over 4 quartiles, each of them containing 25% of the results with a central line in each box representing the median. Apart from the median the mean value is also displayed with a red line slightly sifted to the left in each box. It's important to note that these two values have different interpretations: the median splits the distribution in two equal halves of the data without considering their values,

while the mean indicates the average performance across all trials. **Figure 9** shows the box plots of the two best performing representations: Fragprints and Mordred representations.



(a) MSLL box-plot with fragprints



(b) MSLL box-plot with Mordred descriptors

Figure 9: Box plots of MSLL values for GP models trained on different molecular representations (Fragprints and Mordred), kernels, and training set sizes (1%, 5%, 10%). The central line in each box indicates the median, while the red left-shifted line denotes the mean. These plots highlight performance variability and robustness across multiple randomized trials.

There is one pattern that applies to all the use cases: The smaller the training size, the less precise the surrogate model is. Nevertheless, there are cases where the model with just 1% training samples reaches values near one, meaning that with the ideal selection of the training samples a decent surrogate model can be accomplished.

Another notable aspect across most combinations is the gap between the median and the mean, which indicates the presence of outliers with large MSLL error

values. Just as certain training selections can be highly beneficial for the model, other selections can result in poor performance, which must be avoided. The use of suboptimal surrogate models leads to counterproductive BO, yielding performance that is significantly worse than that of a random search.

Besides using binary kernels, that are applicable in all the considered vector representations, continuous kernels like RBF, Matern and Linear kernel where also tested. Mordred descriptors allow the use of these kernels, as their features are not binary. Most of the considered continuous kernels performed similar to the rest of the other combinations [**Figure 10**], leaving as well the option open to use Mordred descriptors with continuous kernels. An interesting observation regarding the Mordred descriptors is that, for all binary similarity kernels, except for the inner product kernel, the resulting error values were identical.

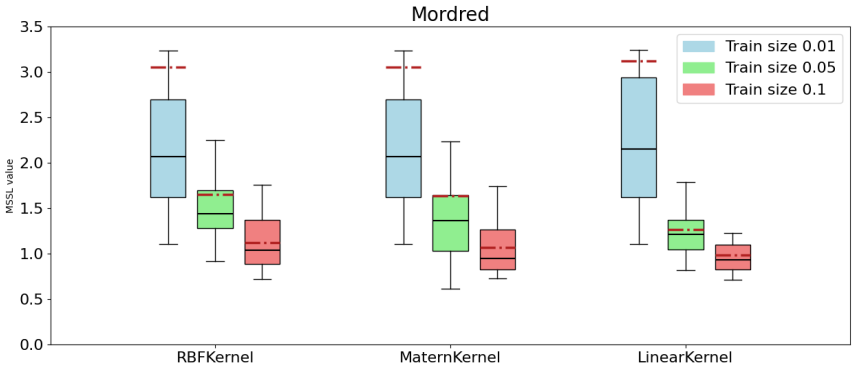


Figure 10: MSLL performance with mordred descriptors and continuous kernels

As previously mentioned, the application of the SSK comes with a high computational cost. This challenge arises first during the calibration and fitting process of the hyperparameters for the GP, and second during the similarity calculation over the held-out candidate set. Even attempts to reduce the kernel's complexity when computing similarities did not sufficiently alleviate these issues. Consequently, despite its theoretical advantages, the full implementation of the SSK was not feasible within the constraints of this work.

Graph representations face a similar problem like the string kernel. The Weisfeiler–Lehman kernel, much like the SSK, computes molecular similarity by evaluating all possible bijections between two graphs and selecting the optimal one to determine similarity. Despite graph kernels computing the similarity between molecules by considering the entire architecture, the generalization error [**Figure 11**] does not improve compared to vector-based representations. Additionally, they come with a higher computational cost, making them less efficient in practice, which becomes particularly problematic when candidate sets grow larger in other scenarios.

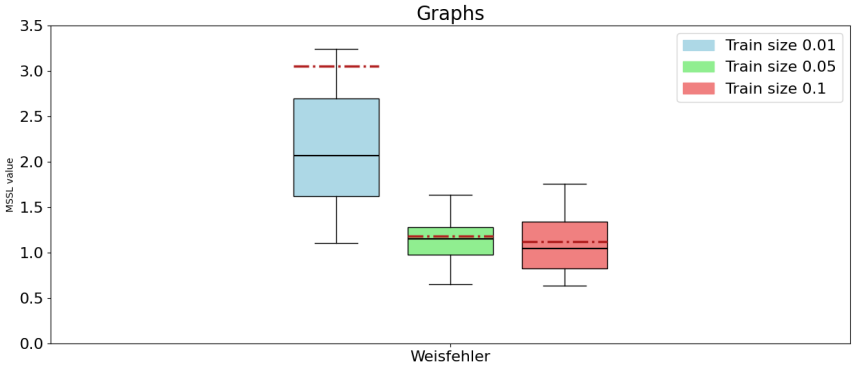


Figure 11: MSLL performance with graph representation and the Weisfeiler-Lehman kernel

Results from **Table 3** indicate that for a small starting set of 1% most of the kernel-representation combinations perform similar, with a high average MSLL error of around three and a very large standard deviation across the different trials. No single combination clearly outperforms the others. The only notable insight is that the 'fragments' representation tends to perform worse than the others, regardless of the kernel used. Additionally, the high standard deviation in MSLL suggests the presence of outliers caused by certain starting sets that result in particularly poor performance; an observation that was initially confirmed through the analysis of boxplots.

Kernel	Molecular Representation				
	ECFP	Fragments	Fragprints	Mordred	Graphs
Tanimoto	3.043 ± 3.70	3.137 ± 3.68	3.049 ± 3.69	3.057 ± 3.69	-
Dice	3.038 ± 3.70	3.580 ± 3.73	3.121 ± 3.67	3.057 ± 3.69	-
Forbes	3.057 ± 3.69	3.580 ± 3.73	3.057 ± 3.69	3.057 ± 3.69	-
InnerProd	3.050 ± 3.70	4.673 ± 4.76	3.064 ± 3.69	3.121 ± 3.68	-
MinMax	3.043 ± 3.70	3.057 ± 3.69	3.043 ± 3.70	3.057 ± 3.69	-
Sorgenfrei	3.049 ± 3.70	4.245 ± 4.45	3.135 ± 3.67	3.057 ± 3.69	-
RBF	-	-	-	3.057 ± 3.69	-
Matern	-	-	-	3.057 ± 3.69	-
Linear	-	-	-	3.121 ± 3.68	-
WL	-	-	-	-	3.057 ± 3.69

Table 3: Mean MSLL errors ± σ for kernel representations using a training size 1%

Increasing the starting set from 1% to 5% makes a significant difference for the GP in the ability to infer a statistical model for the remaining candidates [**Table 4**]. Apart from the expected decrease in the average MSLL error, the standard deviation is drastically reduced. This indicates that poorly performing starting sets, which

would lead to particularly weak GP performance, are largely avoided at this size, with the exception of some cases involving fragment representations. Interestingly, the inner product kernel achieves the best results, particularly when combined with ECFP, Fragprints, and Mordred descriptors. Additionally, graph-based representations using the Weisfeiler–Lehman kernel also obtains a respectable performance.

Kernel	Molecular Representation				
	ECFP	Fragments	Fragprints	Mordred	Graphs
Tanimoto	1.585 ± 0.92	4.24 ± 11.8	1.537 ± 0.90	1.573 ± 0.27	-
Dice	1.447 ± 0.86	1.874 ± 1.46	1.396 ± 0.84	1.573 ± 0.27	-
Forbes	1.564 ± 0.26	1.874 ± 1.46	1.573 ± 0.27	1.573 ± 0.27	-
InnerProd	1.184 ± 0.30	1.672 ± 1.10	1.229 ± 0.54	1.321 ± 0.37	-
MinMax	1.585 ± 0.92	9.073 ± 14.20	1.599 ± 1.06	1.573 ± 0.27	-
Sorgenfrei	1.591 ± 1.06	1.449 ± 0.48	1.485 ± 1.07	1.573 ± 0.27	-
RBF	-	-	-	1.643 ± 0.79	-
Matern	-	-	-	1.624 ± 1.04	-
Linear	-	-	-	1.343 ± 0.38	-
WL	-	-	-	-	1.187 ± 0.36

Table 4: Mean MSLL errors ± σ for kernel representations using a training size of 5%

When 10% of the data is used as the initial set for generating a statistical model of the chemical space, all combinations perform very well [**Table 5**]. The representations that perform best include the Inner Product kernel and the Dice kernel when used with ECFP and Fragprint representations, as well as the graph kernel. Additionally, the use of continuous kernels with Mordred descriptors also yields very strong results.

Kernel	Molecular Representation				
	ECFP	Fragments	Fragprints	Mordred	Graphs
Tanimoto	1.417 ± 0.69	1.305 ± 0.49	1.362 ± 0.68	1.512 ± 0.21	-
Dice	1.141 ± 0.29	1.235 ± 0.42	1.136 ± 0.29	1.512 ± 0.21	-
Forbes	1.480 ± 0.21	1.235 ± 0.42	1.512 ± 0.21	1.512 ± 0.21	-
InnerProd	1.094 ± 0.30	1.150 ± 0.23	1.044 ± 0.35	1.023 ± 0.26	-
MinMax	1.417 ± 0.69	1.386 ± 0.74	1.370 ± 0.68	1.512 ± 0.21	-
Sorgenfrei	1.274 ± 0.47	1.289 ± 0.74	1.254 ± 0.68	1.512 ± 0.21	-
RBF	-	-	-	1.123 ± 0.31	-
Matern	-	-	-	1.069 ± 0.29	-
Linear	-	-	-	1.027 ± 0.25	-
WL	-	-	-	-	1.125 ± 0.36

Table 5: Mean MSLL errors ± σ for kernel representations using a training size of 10%

It is important to keep in mind that the primary goal is to identify the best possible molecule with the fewest observations, as each observation is costly to obtain. In the current scenario, with a total of 405 candidates, starting with a 10% initial set would require 40 random observations before beginning the active BO search. Nevertheless, it is also valuable to understand how different kernel–representation combinations perform when more information is available, as this can provide insights into their robustness and scalability.

In summary, the inner product representation combined with ECFP, Fragprints and Mordred descriptors yeld for all the scenarios the most optimal results. While the use of graph representations with the Weisfeiler–Lehman kernel also produces excellent results, it comes at the cost of higher computational demands.

For the following chapters, the selected kernels will be the Inner Product kernel and the Dice kernel, as they generally outperform other binary kernels. The Tanimoto kernel will also be included, due to its widespread use in molecular similarity tasks, despite not significantly outperforming other kernels in this specific context. Regarding the Mordred descriptors, continuous kernels such as the Matern kernel have shown potential and will be considered. Lastly, graph representations have consistently demonstrated strong performance and remain a promising approach.

5.2 Acquisition function analysis

The selection process in each iteration is implicitly determined by the surrogate model but depends explicitly on the acquisition function. Therefore, the examination of how each acquisition function sets its values for each candidate is an important factor to consider. By observing the acquisition values based on the predicted uncertainty and mean values provided by the GP, it will be possible to track the exploration/exploitation behavior of the BO.

The problem is that, based on previous research, there are currently no established methods or metrics to explain the behavior of acquisition functions. Therefore, it is necessary to develop new approaches that allow the visualization of how different policies operate under varying conditions.

The first method involves computing the mean and variance of the values produced by the acquisition function. By analyzing these two metrics, it becomes possible to deduce the general shape of the acquisition function. A low variance, for example, may indicate a lack of decisiveness, suggesting that most candidates possess similar potential, with none standing out significantly. Plotting these two values over each BO iteration can illustrate how, throughout the process of active search, the acquisition function adapts as new samples are added.

The second method focuses on analyzing the exploitation–exploration balance. Once the statistical model has been obtained from the GP, the BO policy is faced

with two primary options: it can either exploit the existing knowledge by selecting points near regions with high predicted values, or it can explore less certain areas in search of potentially higher values. To track this decision-making process, we compare the predicted value and uncertainty of the selected sample with the average prediction values and uncertainties of all the considered candidates. This allows us to assess how much the selected candidate's values differ from those of the other candidates. In the case of exploitation, we expect the selected sample to exhibit a significantly higher predicted mean value closer to the maximum, whereas in exploration, it is more likely to show a notably higher uncertainty. The degree of separation between the selected sample and the others is quantified by calculating the percentage increase or decrease relative to the averages, providing a clear numerical indication of the strategy being employed **Figure 12**.

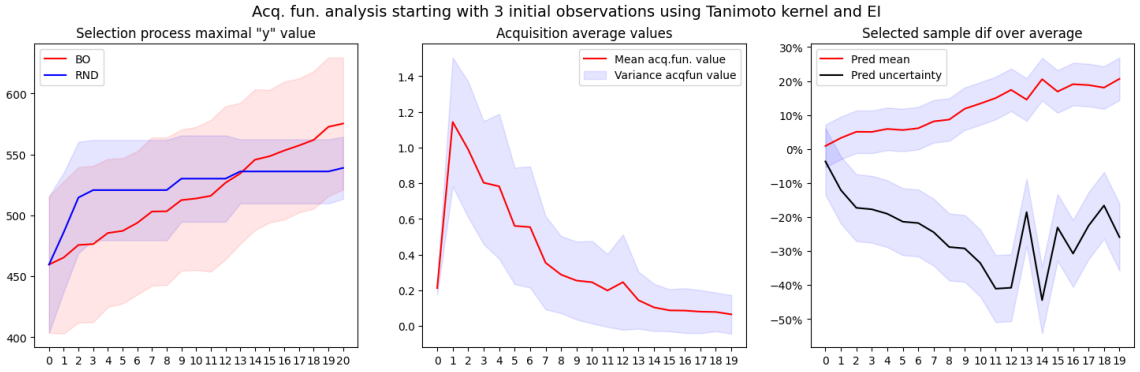


Figure 12: Results using the Tanimoto kernel, molecular fingerprints, and EI with a 1% initial training set, averaged over 10 trials. Left: Best values obtained by BO (red) versus random search (blue) across 20 iterations. Middle: Mean and variance of acquisition function values per iteration, reflecting the function's decisiveness. Right: Deviation of the selected BO sample's predicted mean and uncertainty from the candidate pool, illustrating the exploration–exploitation dynamics.

The main objective is to evaluate the performance of the BO active search while simultaneously tracking the shape of the acquisition function using the first method, and analyzing the exploitation–exploration decision-making with the second. As with the kernel-representation evaluation, results are averaged over 10 independent trials to ensure consistency and reliability.

Using a Tanimoto kernel with a fragprints representation and applying EI as the acquisition function, **Figure 12** (left image) provides an initial illustration of the BO process (in red) compared to random search selection (in blue). This figure visually indicates how BO guides the sampling process at each iteration. Starting with an initial set of three points, the BO process does not show significant improvement over

random selection in the early stages. This is likely due to the acquisition function offering limited variation at the first iteration, making it difficult to identify a clearly promising candidate. However, after the first iteration, the values of the acquisition function begin to spread out, revealing clearer differences in the points considered more promising. Despite this increase in variance, the BO framework tends to favor exploitation, focusing on regions with locally high predicted values and neglecting exploration of new areas. This behavior is evident in the right diagram, where the sampling pattern shows a steady increase in the mean difference, while uncertainty continues to decrease, leading to suboptimal BO performance.

The scenario changes when the BO framework is initialized with a 5% or 10% starting set, showing a clear improvement over the random search **Figure 13**. In both cases, the acquisition function values are widely spread from the beginning. Over successive iterations, the variance decreases until convergence is reached, at which point the BO process stops improving.

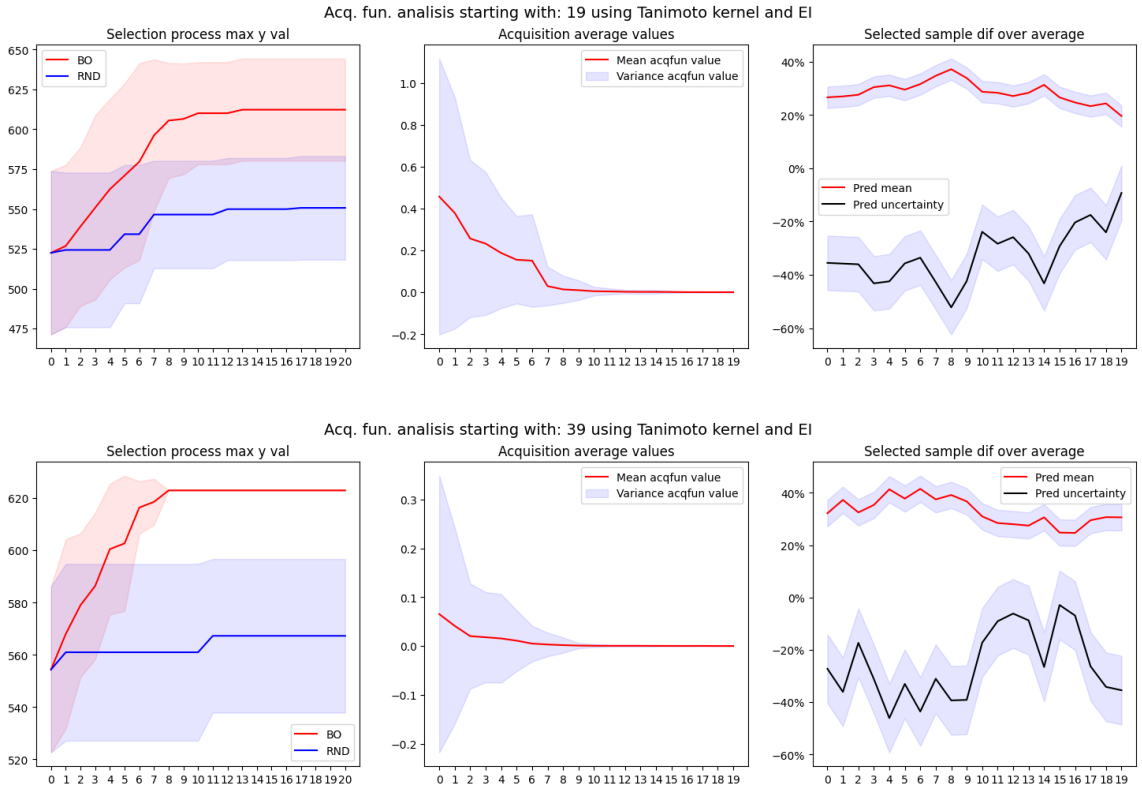


Figure 13: Active search using BO with the Tanimoto kernel, fragprint representation, and EI, using 5% and 10% initial training sets. Left: BO vs. random search performance. Middle: Acquisition function decisiveness. Right: Exploration–exploitation behavior.

With the 5% starting set, once the variance of the acquisition function approaches zero, the framework stops making progress and becomes stuck at the highest acquired value. On the other hand, with a 10% starting set, the BO framework consistently identifies the optimal molecule after 8 iterations in all 10 trials.

The key difference between the two scenarios is that, with a 5% starting set, the framework can form a reasonable picture of the search space but still needs to explore unknown regions to find the optimal value. In contrast, with a 10% starting set, exploration is no longer necessary since there is enough information about the search space to focus solely on exploitation.

If, instead of using EI as the acquisition function within the BO framework, PI is applied, the results and behavior of the selection process, as well as the acquisition function itself, do not show any noticeable differences worth mentioning.

The key finding illustrated by these diagrams is that an initial phase of extensive exploration is crucial for acquiring sufficient information about the search space when starting with small training sets in the BO-Framework. This need for exploration holds true regardless of how the model is constructed for the active search.

One solution to directly address the lack of exploration is to use the UCB as acquisition function and increase the λ factor to encourage exploration. However, using a starting train size of 1% and trying different λ values (0.1, 0.2, 0,5, 1, 5 ,100) the BO framework does not perform better than random selection. Even at a λ value of 100, where the model is clearly forced to prioritize exploration, the BO framework still does not excel beyond random selection, as seen in **Figure 14**. Apart from this fact, one consistent observation across all diagrams using the Tanimoto kernel with a small initial training set of 1% is the lack of variance in the early iterations.

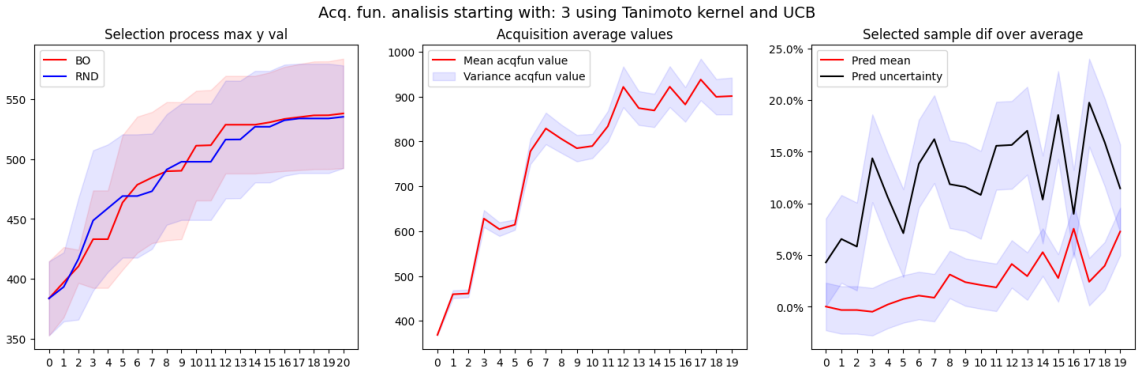


Figure 14: Active search using BO with a 1% initial training set, Tanimoto kernel, and the UCB acquisition function with $\lambda = 100$. Left: BO vs. random search performance. Middle: Acquisition function decisiveness. Right: Exploration–exploitation behavior.

While one might consider acquisition functions that dynamically shift from exploration to exploitation, the persistently poor performance suggests that the limitation lies more fundamentally in the kernel choice. Therefore, instead of further tuning acquisition strategies, a more promising direction is to replace the kernel with one that better captures the structure of the input space.

Changing focus to the inner product kernel, which demonstrated the best overall performance across all applicable representations in the kernel evaluation [**Table 4**] for a 5% starting set. **Figure 15**, equal to the previous images, presents the comparison between BO and random selection, accompanied by the two supporting plots.

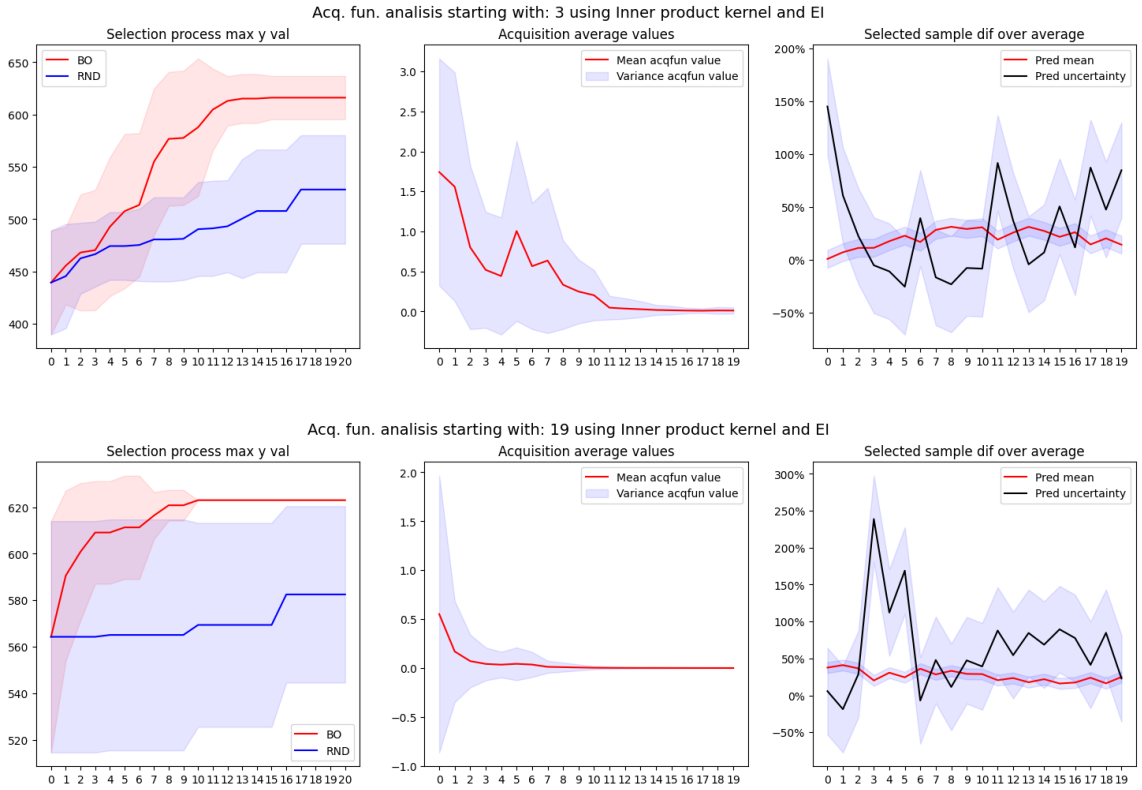


Figure 15: Active search using BO with the inner product kernel, fragprint representation, and EI, using 1% and 5% initial training sets. Left: BO vs. random search performance. Middle: Acquisition function decisiveness. Right: Exploration–exploitation behavior.

The difference is immediately noticeable: while the Tanimoto kernel struggled to outperform random selection, the inner product kernel clearly demonstrates superior performance with just 1% of the data as the starting set, achieving strong

results within only 12 iterations. Furthermore, when BO with the inner product kernel is initialized with a 5% starting set, the framework consistently identifies the optimal molecule within just 10 iterations across all trials. Similar to the Tanimoto kernel, the BO framework stops identifying better molecules once the acquisition variance diminishes. However, the behavior with the inner product kernel is notably different. First, the acquisition variance is initially high during the early iterations and gradually decreases as the optimal molecule is approached. Second, there is a clear and effective balance between exploration and exploitation throughout the iterations. The perfect example of this is observed in the 1% staring set, where in the initial iterations, the framework clearly prioritizes exploration, to later exploit that information, before eventually changing back to explore new regions.

The Mordred descriptor also came as a very promising option when combined with other kernels. However, to apply Mordred descriptors within the BO framework, a dimensionality reduction is required. Unlike other vector representations, Mordred descriptors can posses high continuous values across each dimension, making similarity calculations nearly impossible due to the curse of dimensionality.

To avoid this issue, Principal Component Analysis (PCA) is employed to drastically reduce the dimensionality of the Mordred descriptors. Unfortunately, because of the limited training data available at the early stages of the BO framework, the transformed features are not sufficiently meaningful. This leads to a domino effect throughout the entire active search process, making the approach ineffective for small initial datasets.

One of the final combinations tested within the BO-Framework involves the use of graph representations with the Weisfeiler-Lehman kernel. This approach achieved notably strong results, even when only 5% of the dataset was used as training data. However, when the model is initialized with just 1% of the data, the performance of the BO-Framework becomes comparable to that of random search, offering no significant improvement. In terms of behavior, the model performs as expected during the initial iterations: first exploring uncertain regions to form a rough understanding of the search space, and then exploiting areas with higher predicted values. Despite this, the overall trajectory of the active search closely imitates that of random search, even when 5% of the data is used for training. Even tough the BO-Framework does outperform random search in this setting, the improvement is less distinctive compared to the combination of Innerproduct and Fragprints. Additionally, the use of different acquisition functions had little impact on the performance of the search; and taking into consideration that the optimal strategy is to explore first and exploit later, the UCB would be the most suitable choice for this combination.

In summary, most of the BO setups struggle to perform effectively when provided with very limited data. The notable exception is the combination of fingerprints or fragprints with the inner product, the Sorgenfrei and the Dice kernel, which con-

sistently outperformed random search, even when initialized with only 3 starting samples.

As mentioned in earlier chapters, research related to molecular active search often assumes access to large datasets for training. In cases where BO methods are applied to small datasets, models are typically initialized with a bigger amount of training data, often 5-20% of the available samples [Rai+24; Tom+23; McD+25]. The scenario of starting the model with minimal or no prior information is rarely considered at all. Throughout the initial research, kernels that were initially regarded as the most promising for the task, such as the Tanimoto kernel, were ultimately outperformed by simpler alternatives like the inner product kernel.

The main issue with most of the kernels, regardless of the acquisition function used, lay in the GP model selection, specifically in the process of fitting the GP to find the optimal hyperparameters. As presented in the section **2.1.3**, the loss function that is used to adjust the hyperparameters is the negative marginal log likelihood:

$$-\log p(\mathbf{y} \mid \mathbf{X}, \theta) = \frac{1}{2}\mathbf{y}^\top (K_\theta + \sigma^2 I)^{-1}\mathbf{y} + \frac{1}{2}\log |K_\theta + \sigma^2 I| + \frac{n}{2}\log 2\pi.$$

For most of the cases, minimizing the marginal log likelihood provides suitable hyperparameters for the GP; however, when only a small amount of data is available, the optimizer may converge to a solution where the best explanation for the data is pure noise.

In this situation, the noise variance σ tends to infinity, and the second term of the marginal log likelihood, which penalizes model complexity, grows too slowly (approximately $\approx \frac{n}{2} log(\sigma)$) to compensate the rapid decrease of the first term (scaling roughly as $\approx \frac{1}{2\sigma}$). As a result, the optimizer favors a very high noise level, and simultaneously drives the kernel's outputscale to zero. This leads to degenerate behavior, as seen in **Figure 12** and **Figure 14**, where the acquisition values become uniform during the initial iterations. LOO-CV was also tested to evaluate whether it would mitigate this issue, but the model exhibited the same behavior as with the marginal log-likelihood, showing no improvement.

Returning to the use of the inner product kernel combined with fingerprint/fragprint molecular representations, we propose a two-stage acquisition strategy: first, using UCB with a high initial lambda value to encourage exploration, followed by EI as a polishing step to efficiently identify optimal sample within the most promising regions.

Figure 16 illustrates the results of combining the two acquisition functions to achieve optimal performance. Starting with only 1% of the data as the initial training set, equivalent to just three observations, the BO framework consistently identifies the optimal molecule within 18 iterations across all simulated trials. This means

that, after a total of 21 evaluations, the best possible photoswitch is guaranteed to be found, representing a significant improvement over random search.

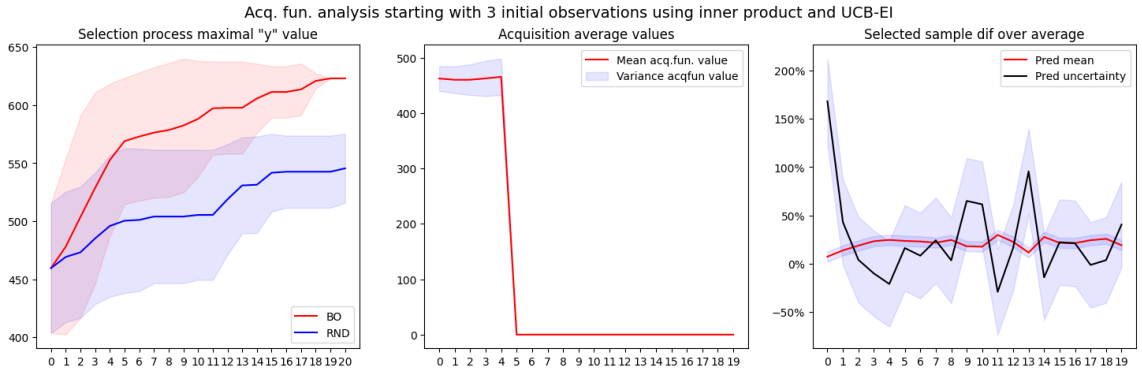


Figure 16: Active search with a 1% initial training set using BO with the inner product kernel and fragprint representation. UCB is applied in the early iterations, followed by EI thereafter. Left: BO vs. random search performance. Middle: Acquisition function decisiveness. Right: Exploration–exploitation behavior.

5.3 Prior selection process

One clear conclusion from the previous section is that initializing the BO-Framework with very little data can be counterproductive for active search. In such cases, the surrogate model may construct an entirely inaccurate statistical representation of the search space, which in the algorithm misguides the BO process. Moreover, when considering a scenario where the BO-Framework starts without a single data point, no model setup proves capable of performing effectively under these conditions.

Since the goal of this research is to identify the optimal molecule with the fewest possible evaluations, relying on an initial random selection for the training set already forfeits the opportunity to optimize the very first few evaluations.

As mentioned earlier, the ideal behavior expected from the BO-Framework is to first explore sufficiently to gain a broad understanding of the search space, and then exploiting the most promising regions. This implies that, in an ideal scenario, the optimal training set would consist of samples that cover the search space as broadly as possible, rather than being concentrated within a single region.

To implement the concept of a prior selection algorithm aimed at identifying the most effective initialization set for the BO framework, three different methods are evaluated to determine their potential in enhancing the active search process. Since the only available information at the initial selection stage is the input features of

the candidate dataset, all applied algorithms fall under the category of unsupervised methods.

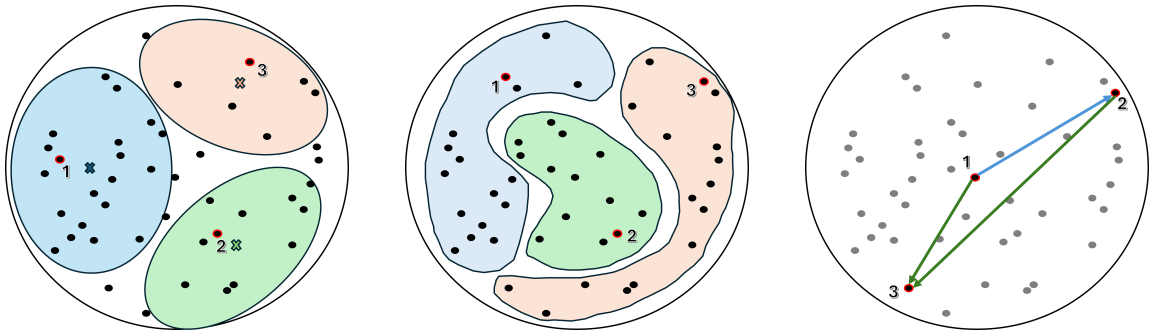
K-means selection: One of the most well-known unsupervised algorithms and clustering methods [SY20]. For the purpose of initialization, the idea is to partition the entire candidate space into n clusters, where n corresponds to the number of initial samples with which the BO framework will begin. After clustering the candidate samples into n groups, each represented by a centroid, the strategy is to select one sample from each group that lies closest to its centroid. This approach aims to produce a diverse and representative initialization set that effectively covers the search space using a limited number of samples.

Spectral clustering: A more sophisticated clustering method compared to k-means. While k-means relies on spherical or elliptical distance metrics to form n clusters, spectral clustering takes a different approach. It does not assume any specific data structure and is capable of identifying clusters with complex, non-convex shapes [BK21]. Given that the distribution of candidates in the high-dimensional search space is unknown, spectral clustering serves as a valuable alternative to k-means. However, unlike k-means, spectral clustering does not produce explicit centroids. As a result, one sample from each cluster is selected at random to form the initial training set.

Least Similarity Sequence (LSS): In addition to the previously discussed clustering methods, a selection strategy is proposed based on the similarity kernels used within the GP framework. The Gram matrix generated by the kernel provides valuable information on the pairwise relationships between all molecules. By utilizing this matrix, the molecule most similar to all others, denoted as x_{center}, is identified, selecting the initial sample that best represents the center of the search space for the selection sequence $\mathcal{X}_{set}$. Subsequent samples are selected iteratively by identifying the molecule that is most dissimilar from both the center and all previously chosen samples. This process is repeated until n samples have been selected, resulting in an initial set defined as $\mathcal{X}_{set} = \{x_{center}, x_{next_1}, ..., x_{next_n}\}$. The goal of this approach is to capture the central region and the outer extremes of the search space, ensuring a diverse and informative initialization.

$$x_{\text{center}} = \max_{x_i \in X} \sum_j k(x_i, x_j) \qquad\qquad x_{\text{next}} = \min_{x_i \in X} \sum_{x_j \in \mathcal{X}_{\text{set}}} k(x_i, x_j)$$

Figure 17 provides a conceptual illustration of how each starting point selection method might behave in the search space. It is important to note that, unlike the simplified visual representation, these algorithms are applied to data in a high-dimensional feature space. Consequently, there is no definitive way to predict which method will perform best, or whether it will improve the performance of the BO selection process compared to a previous random selection.



(a) K-means clustering (b) Spectral clustering (c) LSS

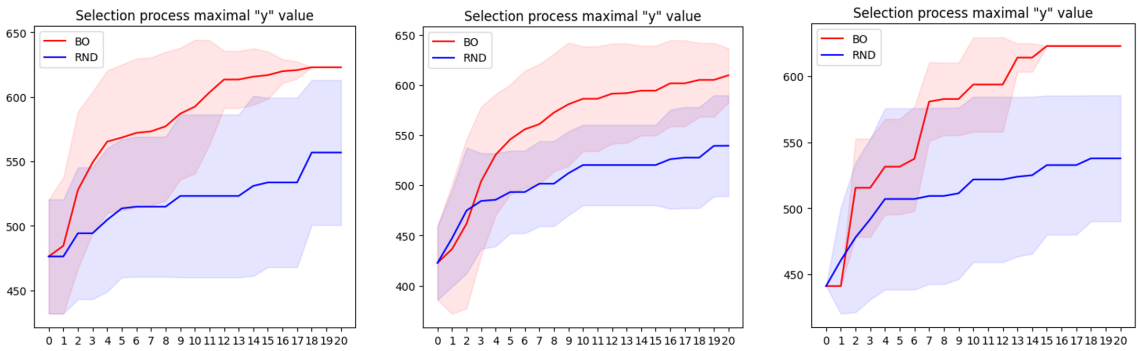
Figure 17: Illustrative comparison of selection algorithms, showing how each method selects 3 samples from the chemical space.

Since the model setup using an inner kernel model with fingerprint/fragprint representation and a combined UCB-EI acquisition function has shown the best performance so far, the proposed prior selection algorithms will be evaluated within this framework. The goal is to achieve faster convergence toward the optimal molecule.

Implementing these prior selection methods on top of the model already reveals different results compared to random selection of starting samples, as shown in fig. 17. The K-means selection method slightly improves performance, achieving full convergence to the optimal molecule one iteration earlier than with random selection. In contrast, the spectral clustering method appears to hinder the active search process. This may be due to the random selection of one sample per cluster, which can still result in a starting set where samples, though from different clusters, are located close to one another, leading to an undesired concentration in a single region of the searched space. Among the tested methods, the least similar sequence strategy delivers the best overall performance. This method consistently identifies the optimal molecule within just 15 iterations across all trials, setting a new benchmark. Under this approach, the BO framework successfully discovers the optimal molecule after only 18 evaluations, including the initial training points selected using the least similarity criterion.

It is also important to note that both the K-means selection and the spectral clustering algorithm contain the effect of randomness. In K-means, the initial centroid is selected randomly from the dataset. Similarly, in spectral clustering, once the data is partitioned into clusters, one representative sample is randomly chosen from each group.

In contrast, the LSS method is entirely deterministic. Its selection process is fully defined from the beginning, ensuring that repeated trials always provide the same output. To evaluate whether this consistent performance is due to an inherently



(a) K-means (b) Spectral clustering (c) LSS

Figure 18: Evaluation of the three proposed initialization strategies: K-means, Spectral clustering and LSS applied to the optimal BO setup.

effective strategy rather than mere coincidence, a controlled function of randomness is introduced into the algorithm. Specifically, instead of always selecting the top-ranked sample, one of the top three candidates will be randomly chosen during each selection step. While this slightly reduces the algorithm's effectiveness, by occasionally preventing the selection of the best possible sample, it reinforces the argument that the overall strategy is robust and not merely reliant on a lucky strike.

That being said, if the LSS method strictly follows its process, by selecting the best candidates according to its defined criteria, the BO-Framework is capable of identifying the optimal molecule after just 13 evaluations, including the initial 3 samples. However, as mentioned earlier, it is not possible to determine with absolute certainty whether this method consistently leads to such an improvement, even with the introduced randomness aimed at validating its strategy. The consistency of this method will be further tested using other datasets in the next sections.

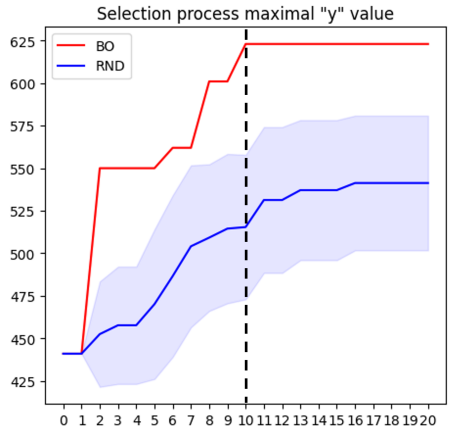


Figure 19: BO with 'Strict' LSS

One final aspect to consider is the number of initial samples selected by the prior selection method that most effectively lead to the optimal value with the fewest evaluations. After testing the model with varying sample sizes from one to ten, the optimal number, consistent with initial tests, was found to be three samples.

5.4 Final model and other datasets

After extensive trials and model adaptations involving different molecular representations, kernels, acquisition functions, and prior selection methods, the optimal setup for identifying the best photoswitch has been established. Among the molecular representations, fingerprints and fragprints delivered the best performance, particularly when combined with the inner product, the Dice and the Sorgenfrei kernels. For the acquisition function, the most effective strategy within the BO framework was a combination of UCB to encourage initial exploration, followed by a switch to EI for refined exploitation. Regarding prior BO set selection, the LSS method, which utilizes kernel-based similarity computations, achieved the best results when initializing with three samples.

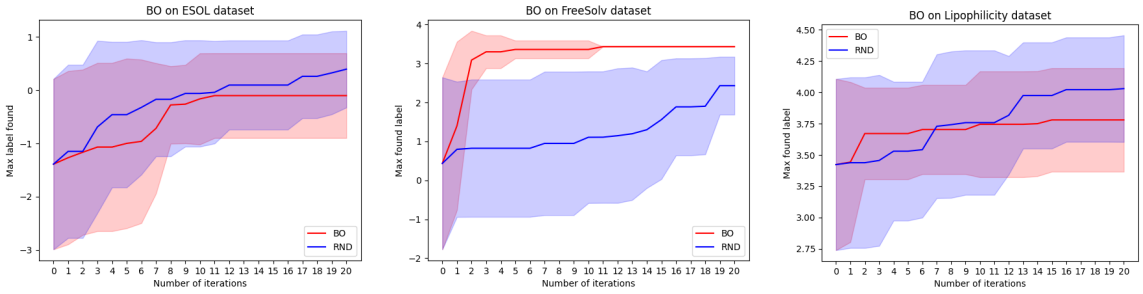
Throughout this research, the BO framework has been developed and refined using only the Photoswitch dataset as a reference. While promising results were achieved for the specific task of identifying the optimal photoswitch molecule, the broader goal is to extend this active search method to other material discovery tasks. To assess how well the developed model performs across different domains, three additional datasets will be tested. The datasets used for evaluation are the following:

- **ESOL**: A dataset containing 1128 compounds, each represented as SMILES strings and labeled with experimentally determined water solubility values. These values range from -11.6 to 1.58, with 1.58 representing the compound with the highest solubility [Del04].
- **FreeSolv**: This dataset comprises 642 molecules, each associated with both experimental and calculated hydration free energies. These energies describe the free energy change when transferring a molecule from an aqueous solution to the gas phase, with values ranging from -25.47 to 3.43 [Riq+18].
- **Lipophilicity**: A dataset containing information on the lipophilic properties of 4200 molecules. The lipophilicity of these compounds is quantified by the partition coefficient, denoted as logP, which ranges from -1.5 to 4.5 [Wan+15].

All of these datasets contain substantially more samples than the Photoswitch dataset, which will likely have a significant impact on model efficiency when computing the acquisition function across all molecules. As with the Photoswitch dataset, the BO framework will be initialized with a starting set using 1% of the data and will be compared against random search. In order to maintain the same scenario, 350 molecules will be chosen randomly from the datasets for testing, which does not impact the overall behavior of the testing process.

However, the results of applying the BO framework as an active search strategy on these larger datasets appear to be disappointing at first sight. While it efficiently

identified the optimal molecule in the Photoswitch dataset after only 13 evaluations, the framework fails to outperform random search on the ESOL and Lipophilicity datasets, both of which contain significantly more samples. The only exception is the FreeSolv dataset, where the BO framework performs comparably well, achieving results similar to those observed on the Photoswitch dataset. Yet, changing the

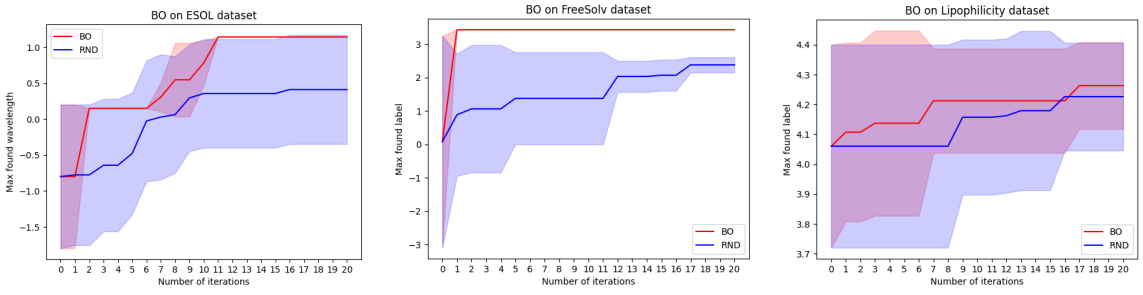


(a) ESOL dataset (b) FreeSolv dataset (c) Lipo. dataset

Figure 20: BO-Framework with the previous defined model

kernel to a more suitable one can dramatically improve the model's performance across certain datasets. For instance, while the inner product kernel struggled with the ESOL dataset, replacing it with the Forbes kernel—without changing any other component of the model, led to a significant performance improvement. However, in the case of the Lipophilicity dataset, switching to alternative kernels within the BO framework did not lead to better results; in fact, performance remained comparable or worse than random search.

Another insight gained from testing the BO framework is the importance of prior selection algorithms. When the kernel performs well in active search, the entire model benefits from using algorithms like the LSS, which heavily rely on the kernel used in the GP model.



(a) ESOL dataset (b) FreeSolv dataset (c) Lipo. dataset

Figure 21: Best achieved BO-Framework for each dataset

To investigate why the BO-based active search fails to perform on the Lipophilicity dataset, the behavior of the acquisition function was analyzed through visualization. However, no clear anomalies or deviations were observed compared to the Photoswitch dataset, despite of this, the model consistently fails to outperform random search.

5.5 Key findings

One of the key early findings in the experimental phase was the efficiency of the linear (inner product) kernel in constructing statistical models with limited data. In contrast, other commonly used kernels, such as the Tanimoto kernel, performed poorly under the same conditions. This is likely due to the binary, high-dimensional nature of the search space, where the inner product kernel is better suited to infer reasonably accurate models from minimal information. As a result, it serves as a strong initial choice for BO when only a small amount of data is available.

Additionally, although graph representations and molecular descriptors initially showed promising results, their application within BO proved less effective. In the case of molecular descriptors, PCA for dimensionality reduction struggled due to the limited number of samples. Similarly, for graph-based representations, the fitting of the GP model using a graph kernel occasionally resulted in erroneous statistical models during certain iterations.

The behavior of the acquisition function was analyzed throughout the iterative active search process using function shape and exploration–exploitation visualization graphs. This analysis showed that the most effective strategy is to begin with broad exploration of the chemical space, followed by focused exploitation of the most promising regions. Specifically, starting with the UCB acquisition function using a high λ value encourages exploration of lesser-known regions. After five iterations, switching to the EI function to exploit the gathered information yielded the best performance across all models.

The biggest challenge throughout the experimental phase was initializing the BO process with very limited data, leading to significant difficulties in fitting the GP model during the initial iterations of the active search. With so few data points available, the model is highly sensitive to the choice of kernel. In some cases, the optimizer may misinterpret the limited information as noise, resulting in inaccurate predictions and disrupting the entire modeling process. To address this issue, employing a prior selection algorithm to choose diverse initial samples proved highly beneficial, helping to establish a more stable foundation for the subsequent BO iterations.

Promising results were achieved in the active molecule search on the Photoswitch dataset. However, applying the same framework to other datasets resulted in un-

expectedly varied performance across tasks, suggesting that dataset-specific characteristics significantly influence the model's effectiveness. For instance, in the case of the ESOL dataset, a simple change of kernel markedly improves performance. In contrast, for the Lipophilicity dataset, the performance remains limited regardless of the model configuration, not achiving better results than a random search model.

For instance, in the ESOL dataset, a relatively simple modification (changing the kernel) resulted in a noticeable improvement in performance, highlighting the importance of kernel selection in certain contexts. In contrast, the Lipophilicity dataset proved more challenging: no model configuration tested was able to consistently outperform a random search strategy. This suggests that certain datasets may inherently constrain the effectiveness of active search strategies, potentially due to greater prediction complexity, limited feature expressiveness, or a weak correlation between the input space and the target property.

These findings underline the necessity of tailoring the modeling strategy to the nature of each dataset, rather than relying on a one-size-fits-all solution.

6 Discussion and Conclusion

This research focused on accelerating the discovery of azobenzene-derived photoswitches by developing a BO-based framework to guide the search process. Once the overall structure of the framework was established, several key components were systematically tested and evaluated, with the primary objective of identifying the most promising photoswitch candidates using the fewest possible evaluations.

Throughout this process, several important insights were gained, particularly in scenarios involving limited data. One of the most critical findings was the significant influence of the kernel choice in the GP model. Kernels such as the inner product, Sorgenfrei, and Forbes showed consistently strong performance under data-scarce conditions, offering more reliable modeling than alternatives like the Tanimoto kernel.

Molecular representation also played a crucial role in the effectiveness of the active search. Vectorial representations, such as extended connectivity fingerprints (ECFP) and fragprints, proved to be not only the most efficient in guiding the search but also the most versatile and computationally inexpensive. These properties made them especially well-suited for integration into the BO-framework.

Regarding acquisition functions, the combination of UCB and EI emerged as the most effective across different configurations. This hybrid strategy consistently achieved superior results, regardless of the underlying kernel or molecular representation. UCB was particularly effective during the initial exploration phase, helping to identify diverse regions of the chemical space, while EI allowed for more focused exploitation in later stages. This balance between exploration and exploitation proved to be a key factor in the framework's overall performance.

In addition, incorporating a prior selection strategy significantly enhanced the efficiency of the framework. The custom-designed LSS algorithm reliably improved performance, provided that the selected kernel was well-suited to the dataset. Similarly, the use of k-means clustering for initial sample selection consistently outperformed random initialization, reinforcing the importance of diverse starting points in BO.

Finally, applying the framework to datasets beyond Photoswitches revealed that strong performance on one dataset does not guarantee success on others, even if the sets are very similar. For example, the framework achieved even better results on the FreeSolv dataset, while on the Lipophilicity dataset, it failed to outperform a simple random search. These variations suggest that the effectiveness of the framework is highly dataset-dependent, influenced by underlying data characteristics that are not always immediately apparent. As such, further investigation is needed to understand how specific dataset features affect the performance of BO-based molecular discovery strategies.

These findings also point to certain limitations and potential directions for future research. One such limitation concerns the use of kernels that are not inherently suited for binary vector representations. In particular, graph kernels underperformed likely due to the relative inefficiency of their current implementation compared to the optimized generation of ECFP features via the RDKit library. Similarly, while a string kernel was explored by generating a Gram similarity matrix, its integration into the BO-framework failed due to execution errors. These were most likely caused by memory limitations or suboptimal implementation. Addressing these computational challenges, through more efficient coding or alternative kernel formulations, could further expand the applicability and robustness of the framework.

Another key challenge that emerged was the limited adaptability of the optimizer and training routines used when fitting the GP and tuning kernel hyperparameters. While the predefined optimization tools provided by BoTorch worked effectively in settings with sufficient training data, they become problematic when the model must be fitted using only a few data points. As discussed in Section 5.2, in data-scarce scenarios, the fitting process often defaults to interpreting the available data as pure noise, or leads the optimizer to degenerate solutions, such as setting kernel parameters to zero. This behavior was particularly evident in kernels like Tanimoto, Weisfeiler-Lehman, and Matern, and likely contributed to the poor performance of several otherwise promising kernel–representation combinations.

A potential solution to this issue would be to introduce priors or bounded constraints on the kernel hyperparameters, thereby preventing the optimizer from converging to non-informative or degenerate values. However, a major challenge with this approach is the lack of guidance on how these bounds should be defined for each parameter across different kernels. Since no information about the data distribution

is available before the initial evaluations, it becomes difficult to establish meaningful parameter intervals with confidence. A potentially more effective alternative would be to implement a modified marginal log-likelihood loss function that includes a penalty term for large noise and output scale values when the number n of training points is small. For instance, incorporating an L2 regularization component could discourage overfitting to noise and guide the model toward more stable parameter estimates in low-data regimes.

The application of the BO-framework to various datasets has also highlighted a critical insight: the choice of kernel can significantly influence performance, depending on the specific dataset and task. This observation opens a valuable direction for future research, establishing practical guidelines or diagnostic procedures to help identify the most suitable kernel based on dataset characteristics, such as data sparsity, feature dimensionality, or molecular representation. Developing such strategies could improve the efficiency and reliability of BO-based frameworks across a wider range of molecular discovery tasks.

In summary, the developed BO-framework demonstrates strong potential for guiding molecular discovery in low-data regimes, with its performance highly dependent on careful choices of kernel, molecular representation, acquisition strategy, and initialization methods. The framework achieved particularly promising results in the discovery of azobenzene-derived photoswitches, as well as on the ESOL and FreeSolv datasets (with the appropriate kernel). However, performance on the Lipophilicity dataset was notably weaker, highlighting the influence of dataset-specific characteristics. These insights provide a useful basis for further refinement and broader application of the framework to diverse molecular discovery tasks, and offer a foundation for greater integration of machine learning in chemical research, particularly in scenarios with limited data availability.

References

[Jac01] Paul Jaccard. "Étude comparative de la distribution florale dans une portion des Alpes et des Jura". In: *Bull Soc Vaudoise Sci Nat* 37 (1901), pp. 547–579.

[Wei88] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *Journal of chemical information and computer sciences* 28.1 (1988), pp. 31–36.

[Mac+98] David JC MacKay et al. "Introduction to Gaussian processes". In: *NATO ASI series F computer and systems sciences* 168 (1998), pp. 133–166.

[Reu99] William Reusch. *Principles of Organic Synthesis*. 1999. URL: `https://www2.chemistry.msu.edu/faculty/reusch/virttxtjml/synth2.htm`.

[Ste99] Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 1999.

[FVB02] Michael A Fligner, Joseph S Verducci, and Paul E Blower. "A modification of the Jaccard–Tanimoto similarity index for diverse selection of chemical compounds using binary strings". In: *Technometrics* 44.2 (2002), pp. 110–119.

[Lod+02] Huma Lodhi et al. "Text classification using string kernels". In: *Journal of machine learning research* 2.Feb (2002), pp. 419–444.

[GFW03] Thomas Gärtner, Peter Flach, and Stefan Wrobel. "On graph kernels: Hardness results and efficient alternatives". In: *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*. Springer. 2003, pp. 129–143.

[Del04] John S Delaney. "ESOL: estimating aqueous solubility directly from molecular structure". In: *Journal of chemical information and computer sciences* 44.3 (2004), pp. 1000–1005.

[Len+04] Thomas Lengauer et al. "Novel technologies for virtual screening". In: *Drug discovery today* 9.1 (2004), pp. 27–34.

[Ral+05] Liva Ralaivola et al. "Graph kernels for chemical informatics". In: *Neural networks* 18.8 (2005), pp. 1093–1110.

[Swa+05] S Joshua Swamidass et al. "Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity". In: *Bioinformatics* 21.suppl_1 (2005), pp. i359–i368.

[XSA05] Huilin Xiong, MNS Swamy, and M Omair Ahmad. "Optimizing the kernel in the empirical feature space". In: *IEEE transactions on neural networks* 16.2 (2005), pp. 460–474.

[Frö+06] Holger Fröhlich et al. "Kernel Functions for Attributed Molecular Graphs–A New Similarity-Based Approach to ADME Prediction in Classification and Regression". In: *QSAR & Combinatorial Science* 25.4 (2006), pp. 317–326.

[McC+06] John McCarthy et al. "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955". In: *AI magazine* 27.4 (2006), pp. 12–12.

[WR06] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[Zha+06] CY Zhao et al. "Application of support vector machine (SVM) for prediction toxic activity of different data sets". In: *Toxicology* 217.2-3 (2006), pp. 105–119.

[Bas08] Igor Baskin. "Fragment descriptors in SAR/QSAR/QSPR studies, molecular similarity analysis and in virtual screening". In: (2008).

[TC08] Roberto Todeschini and Viviana Consonni. *Handbook of molecular descriptors*. John Wiley & Sons, 2008.

[Cut09] Marco Cuturi. "Positive definite kernels in machine learning". In: *arXiv preprint arXiv:0911.5367* (2009).

[CCT+10] Seung-Seok Choi, Sung-Hyuk Cha, Charles C Tappert, et al. "A survey of binary similarity and distance measures". In: *Journal of systemics, cybernetics and informatics* 8.1 (2010), pp. 43–48.

[RH10a] David Rogers and Mathew Hahn. "Extended-connectivity fingerprints". In: *Journal of chemical information and modeling* 50.5 (2010), pp. 742–754.

[RH10b] Maria-Melanie Russew and Stefan Hecht. "Photoswitches: from molecules to materials". In: *Advanced Materials* 22.31 (2010), pp. 3348–3360.

[Xu+10] Qian Xu et al. "Predicting chemical activities from structures by attributed molecular graph classification". In: *2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*. IEEE. 2010, pp. 1–8.

[BB12] H. M. Dhammika Bandara and Shawn C. Burdette. "Photoisomerization in different classes of azobenzene". In: *Chem. Soc. Rev.* 41 (5 2012), pp. 1809–1825. DOI: 10.1039/C1CS15179G.

[Tod+12] Roberto Todeschini et al. "Similarity coefficients for binary chemoinformatics data: overview and extended comparison using simulated and real data sets". In: *Journal of chemical information and modeling* 52.11 (2012), pp. 2884–2901.

[Duv14] David Duvenaud. *Kernel Cookbook*. `https://www.cs.toronto.edu/~duvenaud/cookbook/`. Accessed: 2025-03-08. 2014.

[Sno+15] Jasper Snoek et al. "Scalable bayesian optimization using deep neural networks". In: *International conference on machine learning*. PMLR. 2015, pp. 2171–2180.

[Wan+15] Jian-Bing Wang et al. "*In silico* evaluation of $\log D_{7.4}$ and comparison with other prediction methods". In: *Journal of Chemometrics* 29.7 (2015), pp. 389–398.

[Wij+16] Sony Hartono Wijaya et al. "Finding an appropriate equation to measure similarity between binary vectors: case studies on Indonesian and Japanese herbal medicines". In: *BMC bioinformatics* 17 (2016), pp. 1–19.

[Fab+17] Felix A Faber et al. "Prediction errors of molecular machine learning models lower than hybrid DFT error". In: *Journal of chemical theory and computation* 13.11 (2017), pp. 5255–5264.

[Li+17] Bofang Li et al. "Neural bag-of-ngrams". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[Mor+18] Hirotomo Moriwaki et al. "Mordred: a molecular descriptor calculator". In: *Journal of cheminformatics* 10.1 (2018), p. 4.

[PIT18] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. "Deep reinforcement learning for de novo drug design". In: *Science advances* 4.7 (2018), eaap7885.

[Riq+18] Maximiliano Riquelme et al. "Hydration free energies in the FreeSolv database calculated with polarized iterative Hirshfeld charges". In: *Journal of chemical information and modeling* 58.9 (2018), pp. 1779–1797.

[Wil+18] Liam Wilbraham et al. "High-Throughput Screening Approach for the Optoelectronic Properties of Conjugated Polymers". In: *Journal of Chemical Information and Modeling* 58.12 (2018), pp. 2450–2459. DOI: `10.1021/acs.jcim.8b00256`.

[Yam+18] Tomoki Yamashita et al. "Crystal structure prediction accelerated by Bayesian optimization". In: *Physical Review Materials* 2.1 (2018), p. 013803.

[ZZ18] Mingyan Zhu and Huchen Zhou. “Azobenzene-based small molecular photoswitches for protein modulation”. In: *Organic & biomolecular chemistry* 16.44 (2018), pp. 8434–8445.

[Elt+19] Daniel C Elton et al. “Deep learning for molecular design—a review of the state of the art”. In: *Molecular Systems Design & Engineering* 4.4 (2019), pp. 828–849.

[Pia19] Zbiegniew L Pianowski. *Molecular Photoswitches*. Wiley Online Library, 2019.

[Wu+19] Jia Wu et al. “Hyperparameter optimization for machine learning models based on Bayesian optimization”. In: *Journal of Electronic Science and Technology* 17.1 (2019), pp. 26–40.

[Kre+20] Mario Krenn et al. “Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation”. In: *Machine Learning: Science and Technology* 1.4 (2020), p. 045024.

[KJM20] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. “A survey on graph kernels”. In: *Applied Network Science* 5 (2020), pp. 1–42.

[Mah+20] Batta Mahesh et al. “Machine learning algorithms-a review”. In: *International Journal of Science and Research (IJSR).[Internet]* 9.1 (2020), pp. 381–386.

[Mos+20] Henry Moss et al. “Boss: Bayesian optimization over string spaces”. In: *Advances in neural information processing systems* 33 (2020), pp. 15476–15486.

[SY20] Kristina P Sinaga and Miin-Shen Yang. “Unsupervised K-means clustering algorithm”. In: *IEEE access* 8 (2020), pp. 80716–80727.

[WB20] W Patrick Walters and Regina Barzilay. “Applications of deep learning in molecule generation and molecular property prediction”. In: *Accounts of chemical research* 54.2 (2020), pp. 263–270.

[BK21] Mina Baek and Choongrak Kim. “A review on spectral clustering and stochastic block models”. In: *Journal of the Korean Statistical Society* 50 (2021), pp. 818–831.

[GSC21] David E Graff, Eugene I Shakhnovich, and Connor W Coley. “Accelerating high-throughput virtual screening through molecular pool-based active learning”. In: *Chemical science* 12.22 (2021), pp. 7866–7881.

[BW22] Mickael Binois and Nathan Wycoff. “A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization”. In: *ACM Transactions on Evolutionary Learning and Optimization* 2.2 (2022), pp. 1–26.

[Bur+22] Markus Bursch et al. “Best-practice DFT protocols for basic molecular computational chemistry”. In: *Angewandte Chemie* 134.42 (2022), e202205735.

[Gos22] Arron Gosnell. “A conditional Gaussian process model for molecular property prediction and chemical discovery”. PhD thesis. University of Bath, 2022.

[Gri22] Ryan-Rhys Griffiths. “Applications of Gaussian Processes at Extreme Lengthscales: From Molecules to Black Holes”. PhD thesis. 2022. DOI: 10.17863/CAM.93643. URL: https://www.repository.cam.ac.uk/handle/1810/346223.

[Gri+22] Ryan-Rhys Griffiths et al. “Data-driven discovery of molecular photoswitches with multioutput Gaussian processes”. In: *Chem. Sci.* 13 (45 2022), pp. 13541–13551. DOI: 10.1039/D2SC04306H. URL: http://dx.doi.org/10.1039/D2SC04306H.

[Mül22] Adrian Peter Müller-Deku. “Azobenzene photoswitches: synthetic methodology and biological applications”. PhD thesis. lmu, 2022.

[OHR22] Álmos Orosz, Károly Héberger, and Anita Rácz. “Comparison of descriptor- and fingerprint sets in machine learning models for ADME-Tox targets”. In: *Frontiers in Chemistry* 10 (2022), p. 852893.

[Sta22] StatProofBook. *Conditional Distributions of Multivariate Normal Random Variables*. Accessed: 2025-01-28. 2022. URL: https://statproofbook.github.io/P/mvn-cond.html.

[WGL22] Daniel S Wigh, Jonathan M Goodman, and Alexei A Lapkin. “A review of molecular representation in the age of machine learning”. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 12.5 (2022), e1603.

[Gar23] Roman Garnett. *Bayesian optimization*. Cambridge University Press, 2023.

[Gri+23] Ryan-Rhys Griffiths et al. “GAUCHE: a library for Gaussian processes in chemistry”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 76923–76946.

[JK23] Yimeng Jin and Priyank V Kumar. “Bayesian optimisation for efficient material discovery: a mini review”. In: *Nanoscale* 15.26 (2023), pp. 10975–10984.

[Tom+23] Gary Tom et al. “Calibration and generalizability of probabilistic models on low-data chemical datasets with DIONYSUS”. In: *Digital Discovery* 2.3 (2023), pp. 759–774.

[Jab+24] Kevin Maik Jablonka et al. “Leveraging large language models for predictive chemistry”. In: *Nature Machine Intelligence* 6.2 (2024), pp. 161–169.

[Lu+24] Shuqi Lu et al. “Data-driven quantum chemical property prediction leveraging 3D conformations with Uni-Mol+”. In: *Nature communications* 15.1 (2024), p. 7104.

[Rai+24] Ahmed Shoyeb Raihan et al. “Accelerating material discovery with a threshold-driven hybrid acquisition policy-based Bayesian optimization”. In: *Manufacturing Letters* 41 (2024), pp. 1300–1311.

[Ran+24] Bojana Ranković et al. “Bayesian optimisation for additive screening and yield improvements–beyond one-hot encoding”. In: *Digital Discovery* 3.4 (2024), pp. 654–666.

[BoT25] BoTorch. *BoTorch: A library for Bayesian Optimization in PyTorch*. Accessed: 2025-03-20. 2025. URL: `https://botorch.org/`.

[Dee25] DeepChem. *DeepChem: Open-source library for deep learning in cheminformatics*. `https://deepchem.io/`. Accessed: 2025-03-15. 2025.

[GPy25] GPyTorch. `https://docs.gpytorch.ai/en/stable/index.html`. Accessed: 2025-03-16. 2025.

[Lib25] LibreTexts Chemistry - Williams School. *Representing Molecules*. Accessed: March 12, 2025. 2025. URL: `https://chem.libretexts.org/Courses/Williams_School/Chemistry_I/02%3A_Atoms_Molecules_and_Ions/2.03%3A_Representing_Molecules`.

[McD+25] Matthew A McDonald et al. “Bayesian Optimization over Multiple Experimental Fidelities Accelerates Automated Discovery of Drug Molecules”. In: *ACS Central Science* (2025).

[RDK25] RDKit. *RDKit: Open-source cheminformatics*. `https://www.rdkit.org`. Accessed: 2025-03-15. 2025.