Study on Multi-armed Bandit Algorithm for Sequential Experiments to Predict the Best Molecule with Dynamic Feature Selection

(動的特徴選択による最良分子を推定する逐次実験のための多腕バンディットアルゴリズムの研究)

A Dissertation

Submitted to the Graduate School of Chemical Sciences and Engineering, Hokkaido University, Japan, in Partial Fulfillment of the Requirements of the Degree of Doctor of Philosophy

Md. Menhazul Abedin

Under the supervision of

Professor Tamiki Komatsuzaki

Graduate School of Chemical Sciences and Engineering

Hokkaido University, Japan

August 2024

Declaration

I hereby declare that the matter embodied in this thesis entitled **"Study on Multi-armed Bandit Algorithm for Sequential Experiments to Predict the Best Molecule with Dynamic Feature Selection (**動的特徴選択による最良分子を推定する逐次実験のための多腕バンディットアルゴリズムの研究**)"** is the result of investigations carried out by me under the supervision of **Prof. Tamiki Komatsuzaki** at **Molecule & Life Nonlinear Sciences Laboratory**, Hokkaido University, Japan, and it has not been submitted elsewhere for the award of any degree or diploma. In keeping with the general practice of reporting scientific observations, due acknowledgment has been made whenever the work described has been based on the findings of the other investigators.

Md. Menhazul Abedin

Friday 23rd August, 2024

Dedication

My Parents and Theirs

My Teachers and Theirs

And

Heroes Who Rebuilt Bangladesh Time and Again

Sacrificing Their Lives, Talent, Labor and Devotion.

Acknowledgements

This thesis would not have been accomplished without the encouragement, assistance, and continuous support of many wonderful people. My heartfelt gratitude to all of them. First and foremost, I would like to gratefully thank my supervisor Professor Tamiki Komatsuzaki, for his excellent guidance, encouragement, patience, motivation, cooperation, and financial support during my whole period of the study at Hokkaido University. His wonderful mentorship provided me a great opportunity to be introduced to the field of multidisciplinary research. I am very much thankful for his contributions of incredible ideas and funding to make my PhD research productive and interesting. I want to express my gratitude to Associate Professor Koji Tabata, whose contribution was indispensable in bringing this research to fruition. His expertise on the Bandit algorithm makes me able to work such a challenging domain of research, encourage me to set a new problem definition and motivate me to solve. Yoshihiro Matsumura, I must pay his devotion to this research and fine tuned comments.

I would like to thank my examiners Professor Yasuchika Hasegawa, Professor Tetsuya Taketsugu, Professor Satoshi Maeda and Assistant professor Yuta Mizuno for their very intellectual inputs. I will always remember Pavel Sidorov and Nobuya Tsuji, both are Associate Professors at ICReDD, Hokkaido University, for their constructive suggestions. I would like to express my gratitude to all of my laboratory members, particularly Udoy Sankar Basak, Mohammad

Ali, Md. Mohiuddin, and Professor Md. Al Mehedi Hasan for their significant suggestions and discussions to the success of this research. I am deeply grateful to Hokkaido University for providing me with the opportunity to conduct this research. I am also thankful to Khulna University, Bangladesh, for granting me study leave for the completion of this research. I extend my heartfelt thanks to my wife, child, parents, family members and friends (specially Md. Maniruzzaman) for their unwavering and unconditional support during this challenging journey.

Abstract

My research topic is the acceleration of finding the target molecule whose property (such as drug efficacy, reaction yield, enantioselectivity, wavelength and solvation affinity etc.) is the highest/lowest using a reinforcement learning, linear bandit (Optimism in the Face of Uncertainty Linear Bandit-OFUL bandit) framework. Precisely, the objective of the research is to find the molecule having the highest/lowest quantity of the desired property such as hydration free energy, and reaction enantioselectivity, solvation affinity etc., by fewer experiments as much as possible via bridging chemistry and information science. Usually, chemist do the this job by sequentially testing the molecules which is very time consuming, costly in terms of both money and labor, sometimes it is quite impossible if the set of candidate molecule is very large. To solve this issue sequential optimization is one of the promising approaches in identifying the optimal candidate(s) (molecules, reactants, drugs etc.) with desired properties (such as drug efficacy, reaction yield, enantioselectivity, wavelength and solvation affinity etc.) from a large set of potential candidates, while minimizing the number of experiments is required.

Molecular properties are predicted using fragment-based features (In Silico Design and data Analysis-ISIDA) that are derived from molecular structures represented by Simplified Molecular-Input Line-Entry System (SMILES) codes. However, the high dimensionality of the feature space (e.g., molecular fragment descriptors, Morgan fingerprint descriptors) makes it often difficult to utilize the

relevant features during the process of updating the set of candidates to be examined. In this thesis work, a new sequential optimization algorithm is developed for molecular problems based on a reinforcement learning, multi-armed linear bandit framework, and on-line, dynamic feature selections in which relevant molecular descriptors are updated along with the experiments.

A stopping condition is also designed aimed to guarantee the reliability of the chosen candidate from the pool of data. The developed algorithm was examined by comparing with Bayesian optimization, using two synthetic datasets (termed as Syn-I which is a smaller dataset and Syn-II which is relatively larger dataset) and three real datasets such as a) hydration free energy of molecules, b) energy barrier difference between enantiomer products in chemical reaction and c) photophysical property of molecules e.g., E isomer $\pi - \pi^*$ Transition wavelength. It is found that the dynamic feature selection in representing the desired properties along the experiments provides a better performance (e.g., time required to find the best candidate and stop the experiment) as the overall trend, and that our multi-armed linear bandit approach with dynamic feature selection scheme outperforms the standard Bayesian optimization (BO) with fixed feature variables. The comparison of our algorithm to BO with dynamic feature selection is also addressed.

Contents

List of Figures

Chapter 1

General Introduction

1.1 Motivation of the research

One of the most formidable challenges in chemical science is the search for the optimal molecules that possess desired properties, particularly in drug, material, and catalyst discoveries.[1] The process typically involves a large set of potential candidate molecules, making it a daunting task[2,3] because it require sequential examination each of the molecules. Nowadays size of chemical data is increasing rapidly and getting gigantic in nature. This data explosion is happening in terms of both the chemical substances and their properties. The size of the chemical library or chemical space is getting wider day by day even if it is confined to in a specific domains like medicinal, non-medicinal, organic, or inorganic chemical subspaces[2,3] etc., as for example PubChem contains more than 35 million, ZINC contains over 120 million compounds[4] , CAS registry contains more than 204 million compounds and ChEMBL database is containing close to 2.5 million compounds including 2 million has unique chemical structures.[5] To find a compound/chemicals/molecule that fulfills the requirements e.g., for example

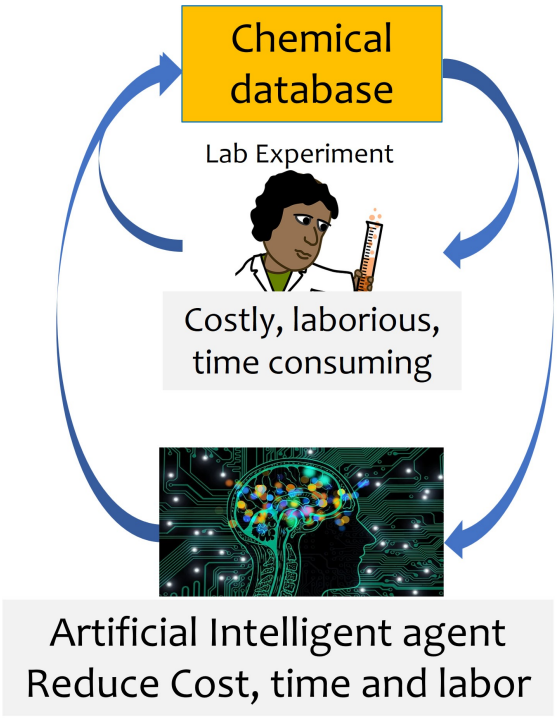

Fig. 1.1: How sequential experiment helps to find the best molecule/candidate in the experimental chemistry is explained in the inner (smaller) part of the figure and the bigger (outer) part explains how the computational method or artificial intelligent agent employed to ease the sequential experiment. (Note: The cartoon are copyright free)

enantioselectivity, solubility, drug efficacy, reaction yield etc., researchers need to test thousands of candidates. For instance, on average, out of 5000 possible candidate drugs, only one drug is expected to be successful.[1] This is very costly task indeed in terms of time, money and labor. Even it is quite impossible for some domain of the research. In this situation some computation approach or artificial intelligent agent are emerged to ease the task of sequential searching of molecules. In computational environment we say sequential searching of molecules as sequential optimization. Using sequential optimization we would like to narrow down the chemical or molecule space and then the best or desired compound/chemicals/molecule entity will be confirmed by experimental chemist. The motivation of the research is explained in the Fig.1.1.

1.2 Sequential optimization for molecular property prediction

Sequential optimization is one of the versatile approaches to solve the sequential experiment problem, whereby candidate molecules are iteratively tested for synthesis based on the predictions of an adaptively updated machine learning model.[1,6,7] The primary objective of this approach is to identify optimal molecules with desired properties while minimizing the number of experiments required. The recent advancements of robotics in chemistry have been further stimulated research in sequential optimization algorithms, making it a promising direction for future study.[8,9,10] One of the most widely used methods for solving sequential optimization problems in chemistry is Bayesian optimization (BO).[11,12,13,14,15,16,17] This method involves constructing a probabilistic model that maps chemical conditions, such as reaction conditions, catalysts, and solvents, to the property of interest. Importantly, the sequential strategy used to propose the next candidate balancing the trade-off between exploring new chemical conditions and exploiting known high-performing conditions.[18] The probabilistic model is updated with new data as it becomes available, allowing the algorithm to iteratively search for the optimal molecule. This approach can reduce the number of experiments required, thereby saving time and resources, particularly for the exploration of complex conditions that are difficult to navigate using traditional trial-and-error approaches.[19,20,21]

However, the high dimensionality of features required to describe molecules causes a significant challenge for the application of this method in chemistry, as in the Gaussian process regression model. This challenge can be addressed through the use of various techniques, such as feature selection or introduction

of regularization parameter(s), and so on. For example, the feature selection based on the Automatic Relevance Determination (ARD) such as SAASBO[22] can be utilized for solving the high dimensionality problem. However, the regression can easily lead to over-fitting in small data set, which indicates the difficulty in applying to chemical problems.[16] Additionally, the construction of stopping conditions for identifying optimal molecules remains an important issue, as the optimal point is often unknown until all candidate molecules have been examined.

Another approach for solving sequential optimization problems is multi-armed bandit (MAB) algorithm, which could be regarded as a discrete version of BO. BO uses Gaussian processes as likelihood functions, including traditional homoscedastic models and more sophisticated heteroscedastic models where the noise level varies with the input.[23,24,25] Additionally, to handle complex data distributions, transformations such as the Box-Cox transformation[26] or normalizing flows[27] are also employed. In contrast to these BO approaches, the most general class of MAB algorithm does not necessarily require to define neither a model nor smooth continuity along underlying feature variable(s) a priori, but solely to assume sampling from identically independent distribution (i.i.d.) whose shape is unknown.[18,28,29,30,31,32] For example, best arm identification problem of MAB algorithm[33,34,35,36,37] is aimed at choosing the arm (attached to a slot machine among many slot machines in front of a person) whose expected rewards (e.g., number of coins) is the largest with as fewer number of pulling arms of slot machines as possible. Note that the output of rewards is probabilistic for each arm and only at the limit of infinite number of pulling each arm one can precisely know what expected rewards (i.e., true mean) is for each arm. The mathematical aspects of the bandit algorithms have been well

studied, giving it an advantage in the construction of stopping conditions for finding the best slot machine. In recent years, this algorithm has been applied to chemical problems, such as drug discovery, where each drug candidate is considered as an arm attached to each slot machine, and its efficacy as a drug is evaluated.[6,7,38,39] However, the direct application of the good arm identification bandit problem to chemical problems is not necessarily the most appropriate, as chemical properties are observed experimentally at once or few for each condition in most cases, that is, not so often like several hundreds to more for each experimental condition. This is because macroscopic experiments are considered as less stochastic (or rather deterministic) than single molecule experiments. Note however that the experimental values include measurement (aleatoric) noise of laboratory apparatus and depend on the level of skill of experimentalists, which are considered to be compensated by introducing a noise term into the regression models.

There exist several variants of good arm identification problems in MAB algorithms such as "Gaussian Process-bandits[40,41] (corresponding to BO)", "threshold bandits[42,43]", "bad arm identification problem[30]", "classification bandits[44,45]", and so forth. In the linear bandit, instead of regarding "molecule entity" as an arm to generate a reward (i.e., observed property associated with each molecule), the reward is represented by a multilinear regression that is a linear summation of some functions of feature variables.[31] The linear model derived from the observed rewards is subsequently applied to predict the rewards of unobserved candidate molecules. This prediction aids in suggesting the next molecule to be tested. Linear bandit algorithms offer several advantages, including low computational costs and ease of implementation on large datasets. However, the performance of linear bandit algorithms requires further investigation in chem-

ical problems where the dimensionality of chemical conditions is often larger than the available data points.

To overcome the challenges associated with the high dimensionality of features of molecules, we have developed a linear bandit algorithm that incorporates dynamic feature selection scheme, i.e., the relevant feature variables are dynamically updated along the accumulation of the data to be examine. In particular, we have theoretically constructed a stopping condition within this framework. To evaluate the performance of our approach, we compared linear bandit algorithms, specifically Optimism in the Face of Uncertainty Linear (OFUL) bandit with and without dynamic feature selection, Bayesian optimization (BO) with and without dynamic feature selection, and random search (RS) using synthetic data and realistic chemical optimization problems of solubility,[46] enantioselectivity[47] and photoswitch.[48,49]

1.3 Tools used in this study

The Python programming language is utilized for data analysis, while ChemDraw 19.0 is employed to draw molecular structures from simplified molecular-input line-entry system (SMILES) codes. Microsoft Office is used for presentations, and the thesis is written on the LaTeX platform (Overleaf).

1.4 Layout of the thesis

This thesis have been composed in seven chapter as follows. **Chapter 1** is the the introductory chapter where I explain the motivation and purpose of the study. In **Chapter 2** I explain the methods for the sequential optimization. This chapter includes development of the bandit algorithm. **Chapter 3** presents the results of

synthetic data along with how the datasets are generated (Synthetic-I data and Synthetic-II data). **Chapter 4** presents the results of sequential optimization for free solvation energy (FreeSolv) data. **Chapter 5** presents the results of sequential optimization for experimental enantioselectivity data. **Chapter 6** presents the results of sequential optimization for photoswitch data. And finally **Chapter 7** presents the general conclusion of the thesis work.

Chapter 2

Theoretical Developments on Sequential Optimization

In the advancement of current world enormous problems have been solved and hidden truth of the nature is revealed by Machine Leaning (ML) methods ranges from social science, literature to business and all branches of science. Machine learning has been changed the view of business policy, medical treatment strategy, and have been given ease condition in research and education etc.

But what exactly is machine learning? Simply Machine learning is a computer-based approach which can utilize available data to make decision or prediction for unknown event. Formally, Machine learning represents a ground breaking domain within the broader realm of artificial intelligence that empowers computers to learn and make predictions or decisions without explicit programming. It revolves around the development of algorithms that enable machines to recognize patterns, infer insights, and improve their performance over time through experience and data analysis. The essence of machine learning lies in its ability to autonomously adapt and evolve, making it a key driver in solving complex

problems across various domains, such as healthcare, finance, and technology. By leveraging statistical techniques and computational power, machine learning has the potential to revolutionize decision-making processes and unlock new possibilities in the way we approach problem-solving and data interpretation. Machine learning can be broadly classified into three main categories based on the nature of the learning and the type of feedback or supervision provided to the algorithm:

- **Supervised learning:** Supervised learning undergo training a labeled dataset, where each input (independent variable) is paired with output (some literature say target, dependent variable). The goal of the algorithm is to learn a mapping function from the input to the output, and then making predictions or classifications on new, unseen data. Very common applications of supervised learning includes image recognition, speech recognition, and spam filtering etc.

- **Unsupervised learning:** Unsupervised learning involves training the algorithm on unlabeled data, without explicit guidance on the desired output (target or dependent variable). The algorithm seeks to identify patterns, relationships, or structures within the data. Clustering and dimensionality reduction are common tasks in unsupervised learning. Examples include customer segmentation, anomaly detection, topic modeling, dimensionality reduction and clustering etc.

- **Reinforcement learning:** Reinforcement learning involves an agent learning to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or punishments based on the actions it takes. The objective is for the agent to learn a strategy or pol-

icy that maximizes cumulative rewards over time (alternatively minimize cumulative regret). Applications of reinforcement learning include game playing (e.g., AlphaGo), robotic control, and autonomous systems etc.

These classifications provide a comprehensive framework, and within each category, numerous algorithms and techniques are designed to address specific tasks and challenges. In this research we merely would like to study **Reinforcement Learning** framework for sequential optimization problem in chemistry. There are numerous reinforcement learning methods exists in literature. Very well known methods are: Multi-armed bandit, Markov decision process, dynamic programming, SARSA (State-Action-Reward-State-Action), Q-learning, Monte Carlo Methods etc. We are confined here on Multi-armed bandit algorithm.

2.1 Introduction to Bandit algorithm

The bandit algorithm is a widely used reinforcement learning method designed to address sequential decision-making problems. The origins of bandit problems can be traced back to the fields of sequential decision-making and optimization. This framework deals with making decisions under uncertainty and was initially introduced by William R. Thompson in a 1933 article published in Biometrika, where he proposed adaptive treatment allocation rather than a random approach. Approximately two decades later, in the 1950s, Frederick Mosteller and Robert Bush coined the term "Two-Arm Bandit," describing a scenario where one can choose between two options (or "arms"), each offering a reward from an unknown distribution. The term "bandit" is inspired by the concept of a slot machine, or "one-armed bandit," where a gambler must decide which machine to

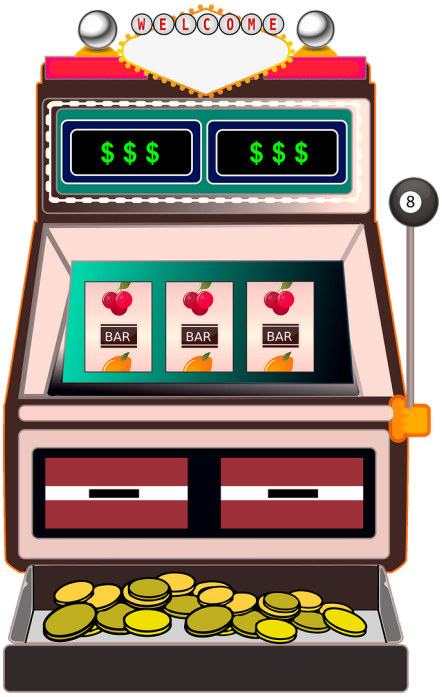

Fig. 2.1: Slot machine (Photo source: https://pixabay.com/)

play to maximize their cumulative rewards.

Use of Bandit algorithm is getting more attention in the scientific community of clinical trial, drug discoveries, chemical optimizations, chemical synthesis, product predictions and so on. This method is currently extensively used in technology companies for web configuration, news recommendation, dynamic pricing, ad placement and many more. Examples of Bandit algorithm are includes Epsilon-Greedy Algorithm, UCB1 (Upper Confidence Bound), Thompson Sampling, Exp3 (Exponential Weighted Exploration and Exploitation), Softmax Exploration, Bayesian Bandits, KL-UCB (Kullback-Leibler Upper Confidence Bound), Contextual Bandits, LinUCB etc.

2.1.1 How does bandit algorithm works?

The term "bandit" is borrowed from the concept of pirates who take money from their victims, similar to how a slot machine extracts money from players. To understand the algorithm intuitively, refer to Fig.2.2. Imagine a player standing in front of several slot machines, each with an unknown probability of winning (although the dealer might know these probabilities). The player earns a reward of 1 for winning coins and 0 otherwise. The reward denoted as R. In this example R is binary but continues reward is frequently considered in real problem. The player expect to maximize his earning from this game. Again, further R_t might be used for reward, which means reward at t^{th} play or reward at time t. In real application reward (R) is the molecular property.

Initially, the player tests the first slot machine and wins coins. Next, the player tries the second slot machine and loses. With this information—that the first machine was successful and the second was not—the player must decide the next move. The player has two options: a) continue playing the machines already tested, or b) explore other machines to gain additional information.

The player then chooses to play the first machine again and wins coins once more. At this point, the player aims to identify the best slot machine. While it might seem that the first machine has a high chance of being the best, there is a risk of missing out on a potentially better machine if the player does not explore the other machines. Balancing between exploiting the known successful machine and exploring new options is crucial.

In reality, examining the underlying distribution of the slot machines reveals that the third slot machine is actually the best, but the player has not yet tried it. This presents a dilemma: how should the player proceed?

The key to this algorithm is finding a balance between exploiting known

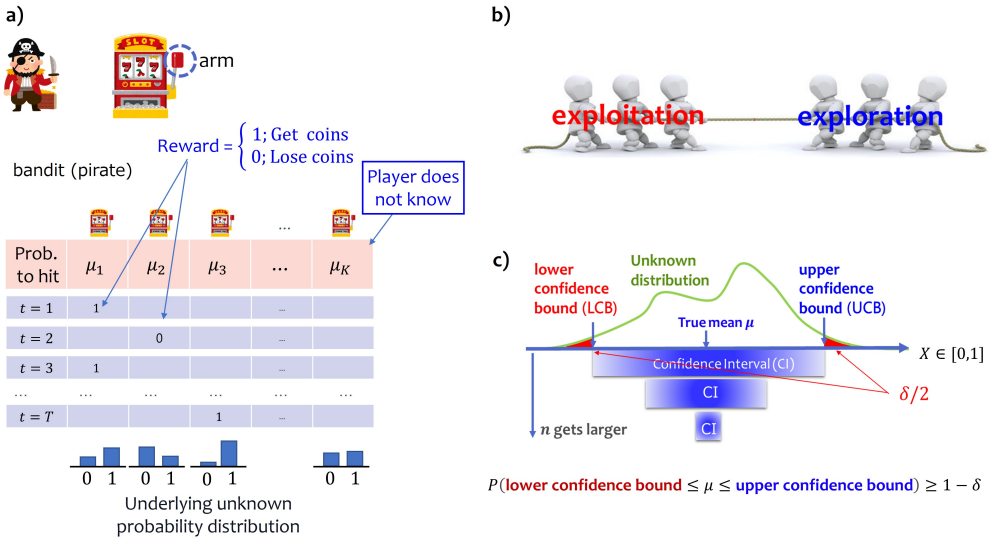

Fig. 2.2: a) How slot machine example used to develop multi-armed bandit algorithm is explain, b) How to balance exploration and exploitation, and c) how the sample mean converge to true mean using confidence interval.

information and exploring new possibilities. This balance is achieved using a concept called the confidence interval. A confidence interval provides a range within which the true mean of an unknown distribution is likely to lie, given a specified level of confidence. Although the true distribution of rewards is unknown, the confidence interval offers a range where the true mean is expected to fall, allowing for some margin of error.

As the player continues to test the slot machines and the sample size (n) increases, the confidence interval becomes narrower. This narrowing indicates that the sample mean is converging towards the true mean of the distribution. The confidence interval can be analytically derived under certain assumptions, such as rewards being Independently and Identically Distributed (IID). This approach helps the player make informed decisions, balancing between exploiting known successful options and exploring potentially better alternatives.

2.1.2 Terminologies of bandit algorithm

The bandit problem is a sequential game played between an agent and an environment. This game typically unfolds over a set number of rounds, referred to as the horizon (T), which is a positive real number. During each round or iteration step $t \in [T]$, the agent selects an arm a_t from a set of arms K, and the environment provides a reward R_t from an unknown distribution. The goal of the agent is to maximize the total reward:

$$\max\left(\sum_{t}^{n} R_t\right).$$

To achieve this, we assume the player has no knowledge of future plays, and the choice of the current arm is based solely on previous history $(a_1, a_2, ..., a_{t-1})$. The following terminologies are commonly used in the context of the bandit problem:

- **Arm:** Represents the actions available to the player. In this research, an arm corresponds to different molecules in the feature space.

- **Bandits:** A collection of arms. When multiple options are available, this collection is called a Multi-armed Bandit. In this context, the collection of molecules to be tested constitutes the Bandits.

- **Agent and Environment:** The agent is the learner and decision-maker in the bandit problem. Anything other than the agent is referred to as the environment. The environment interacts continuously with the agent, who selects actions based on the environment's feedback and adapts accordingly for subsequent actions.

- **Policy:** A policy is a mapping from the agent's past actions to the actions

the agent should take. The agent develops a policy based on its interactions with the environment to maximize rewards.

- **Value function:** This function indicates what is beneficial in the long run. Essentially, the value of a state is the total expected reward an agent can accumulate from that state onward.
- **Reward:** The quantity received by the agent after taking an action. In chemical synthesis, this might refer to chemical properties, while in drug screening, it could indicate drug efficacy.
- **Regret:** Regret measures the loss incurred by the agent. It is the difference between the rewards of the optimal arm and the sub optimal arm suggested by the algorithm.
- **Horizon:** The number of trials remaining in the game.
- **Exploitation:** An algorithm exploits when it selects the arm with the highest estimated value based on past plays.
- **Exploration:** An algorithm explores when it chooses an arm that does not have the highest estimated value based on past plays. In other words, exploration occurs whenever exploitation does not.
- **Exploration-exploitation trade off:** This is the balance that any learning system must maintain between the desire to explore new options and the desire to exploit known, rewarding options.

2.1.3 Traditional bandit algorithm

To understand how the Bandit algorithm works, let's briefly explore two traditional bandit algorithm. Consider a scenario where we repeatedly choose an

option or action from a set K, which provides a numerical reward from an unknown distribution. The goal is to maximize the expected total reward over a predefined horizon T. This situation represents the basic K-Bandit problem. In our K-armed bandit problem, each arm a has an expected or mean reward when selected; this is known as the value of that arm. We denote the arm chosen at time step t as a_t and the corresponding reward as R_t. The true value of any action a, denoted $Q(a)$, is the expected reward when a is selected:

$$Q(a) = [R_t \mid a] \tag{2.1}$$

If we knew the values of $Q(a)$ for all arms, we could always select the arm with the highest value. However, these values are not known beforehand (can only be access for all data). Now the question is how to find this values. we estimate the value of arm a at time step t as $q(a)$ which is sample estimate. Here we assume $q(a)$ will be converse to $Q(a)$ for infinite trial.

We may observe that at least one action will have the highest estimated value; these are called greedy actions, while those with smaller estimated values are non-greedy actions. Selecting one of these greedy actions means exploiting current knowledge of action values. If, instead, a non-greedy action is chosen, it is called exploring, as it helps improve the estimate of the non-greedy action's value. Exploitation is optimal for maximizing the expected reward in the short term, but exploration might lead to a greater total reward in the long run.

The simplest action selection rule, the Action Value Method, involves selecting one of the actions with the highest estimated value. If there is more than one greedy action, a selection is made among them arbitrarily, perhaps randomly. This greedy action selection method can be expressed as:

$$a_t = \arg\max_a q_t(a) \tag{2.2}$$

where argmax denotes the action a for which the expression is maximized.

Up to this point, we haven't incorporated exploration, which is necessary due to the uncertainty in the accuracy of the action value estimate $q_t(a)$. The Action Value Method always selects the best arm based on current estimates, but other arms might actually be better. Therefore, it is meaningful to sometimes select from the non-greedy actions. The Upper Confidence Bound Method introduces exploration by adjusting the arm selection policy as follows:

$$a_t = \arg\max_a \left[q_t(a) + c\sqrt{\frac{ln\ t}{N_t(a)}} \right], \tag{2.3}$$

where $ln\ t$ is the natural logarithm of time step t, $N_t(a)$ is the number of times arm a has been selected prior to step t, c controls the exploration rate, and $\sqrt{\frac{ln\ t}{N_t(a)}}$ measures the uncertainty of the arm a. This equation anticipate to chose the arm from the non greedy set as well. This problem structure is very basic. In this thesis we have used Optimism in the face of uncertainty linear (OFUL) bandit explain in the next section where we also use Upper Confidence Bound (UCB) policy but the computation strategy is different.

2.1.4 Optimism in the face of uncertainty linear (OFUL) bandit

Up to this point, the contexts or associated information of the arms have not been considered. However, understanding this information is crucial for learning about the arms. For example, in the personalized advertisement problem, the

suggestion of a product may depend on factors like location, age, and gender of an internet user. When additional or contextual information is incorporated into the Bandit problem, it is known as a **contextual bandit**. This added complexity makes the model more applicable to real-world scenarios. Consequently, bandit methods have been fundamental in the principles of reinforcement learning, contributing to the development of algorithms capable of tackling more complex decision-making problems in dynamic environments.

In a standard multi-armed bandit problem, an agent chooses between multiple actions (arms) without any extra information. In a contextual bandit problem, each arm is associated with additional information or context, which the agent receives before making a decision. The objective is to learn a policy that maps each context to the most suitable action, maximizing the cumulative reward over time.

Contextual bandit algorithms have applications in various fields, such as online advertising, recommendation systems, and personalized content delivery, where decisions must be tailored to the specific context of the user or situation.

In this research, a contextual bandit algorithm is being developed for sequential experiments involving molecular properties, using topological features as the context. The developed algorithm, known as the Optimism in the Face of Uncertainty Linear (OFUL) bandit algorithm, is explained in the following sections.

The OFUL bandit[31] algorithm assumes the following linear regression for the molecular property in question (called "reward" hereinafter) y_i of the i^{th} molecule,

$$y_i = \theta_*^T f(x_i) + \eta_i \tag{2.4}$$

where x_i denotes the vector representing a set of the molecular features asso-

ciated with the i^{th} molecule, that is, $x_i := \{x_{i,k}\}$ $(i = 1,\ldots,N_{\text{tot}}, k = 1,\ldots,N_f)$. Here, $x_{i,k}$ denotes its k^{th} feature variable of the i^{th} molecule, and N_{tot} and N_f are arbitrarily large total number of candidate molecules and the total number of the feature variables, respectively. f denotes an N_f set of arbitrary functions composed of x_i, that is, $f(x_i) := \{f_k(x_i)\}$ $(k = 1,\ldots,N_f)$. θ_* corresponds to the N_f-dimensional vector of weight parameters θ_{*k} assigned to each $f_k(x_i)$, which can be acquired only using the full N_{tot} set of the candidates, and is unknown in general when smaller set of molecules is only available. η_i characterizes some deviation of the regression model from the actual reward, modelled by R-sub-Gaussian noise. The R-sub-Gaussian distribution[50] corresponds to a Gaussian distribution with zero mean, $\mathbb{E}[\eta_i] = 0$, and variance bounded by R^2, $\mathbb{E}[\eta_i^2] \leq R^2$. In this research, an identical function $f_k(x_i) = x_{i,k}$ was simply employed.

In this thesis, the values of the weight parameters θ are estimated along the iteration step t of the algorithm, denoted by $\hat{\theta}_t$, which may differ from the underlying unknown true θ_*. An l^2-regularized least-squares estimate of θ_* at iteration step t is provided by[31]

$$\hat{\theta}_t = (X_{1:n_t}^T X_{1:n_t} + \lambda I)^{-1} X_{1:n_t}^T y_{1:n_t} \tag{2.5}$$

where $n_t = t + 10$ because we begin with 10 randomly chosen molecules at the initial step $t = 0$, $\lambda(> 0)$ is the regularization parameter and I is identity matrix of size $n_t \times n_t$. $X_{1:n_t}$ is the matrix represented by $\left(x_{i_1}^T, x_{i_2}^T, ..., x_{i_{n_t}}^T\right)$ where the indexes from i_1 to i_{10} represent the indexes of molecules randomly chosen initially and i_{11} to i_{n_t} those of molecules "observed" up to iteration step t, and the vector $y_{1:n_t} = (y_{i_1}, y_{i_2}, ..., y_{i_{n_t}})^T$ whose elements are observed rewards up to step t.

The confidence region (termed as confidence ellipsoid[31]) that includes true

weight parameters θ_* with high probability at least $1-\delta$ can be constructed using the estimated parameters $\hat{\theta}_t$ as follows:[31]

$$C_t = \left\{ \theta \in \mathbb{R}^{N_f} : ||\theta - \hat{\theta}_t||_{\overline{V}_t} \leq R\sqrt{2\log\left(\frac{\det(\overline{V}_t)^{1/2}\det(\lambda I)^{-1/2}}{\delta}\right)} + \lambda^{\frac{1}{2}} S \right\}, \tag{2.6}$$

where $||\theta - \hat{\theta}_t||_{\overline{V}_t} = \sqrt{(\theta - \hat{\theta}_t)^T \overline{V}_t (\theta - \hat{\theta}_t)}$ for all $t \geq 0$. Here, δ is a user-controllable parameter for failure probability and another parameter S is chosen so that the upper bound of θ_* is lower than or equal to the S. The matrix $\overline{V}_t$ is defined as $\lambda I + \sum_{m=1}^{n_t} x_{i_m} x_{i_m}^T$. In this instance, 0.05 is used as a typical value for the allowance-error rate δ.

The Optimism in the face of uncertainty (OFU) principle suggests the next candidate (or molecule), denoted by $x_{i_{t+1}}$, using the following equation:

$$(x_{i_{t+1}}, \tilde{\theta}_{i_{t+1}}) = \underset{(x,\theta)\in D_{t+1}\times C_t}{\arg\max} \; \theta^T x. \tag{2.7}$$

These two vectors, $x_{i_{t+1}}$ and $\tilde{\theta} i_{t+1}$, are chosen to maximize the expected reward $\theta^T x$ among the unobserved remaining candidates at iteration step t (D_{t+1}). Each candidate is characterized by its molecular feature vector x (denoted by $x \in D_{t+1}$) within the confidence region (C_t) of the weight parameters θ (as described in Eq. 2.6). This approach balances exploitation and exploration, similar to the upper confidence bound strategy, which is effective in efficiently finding the maximum reward.[11,12,51] It is important to note that $\tilde{\theta} i_{t+1}$ is different from the evaluated parameter $\hat{\theta}_{i_{t+1}}$.

2.1.4.1 Computational details

In this study, the analytical solution of the upper bound of expected reward for each candidate whose molecular feature vector is represented by $x(=\{x_k\})$ $(k=1,\ldots,N_f)$ is computed using Eqs. (2.6) and (2.7). The solution is expressed as

$$\begin{aligned} \max_{\theta\in C_t} \theta^T x &= \hat{\theta}_t^T x + \tilde{B}||x||_{\overline{V}_t^{-1}}, \\ \tilde{B} &:= R\sqrt{2\log\left(\frac{\det(\overline{V}_t)^{1/2}\det(\lambda I)^{-1/2}}{\delta}\right)} + \lambda^{\frac{1}{2}} S \end{aligned} \tag{2.8}$$

This formula can be derived by Lagrangian multiplier method for maximizing $\theta^T x$ under the constraint, $||\hat{\theta}_t - \theta||_{\overline{V}_t} = \tilde{B}$ using the fact that the expected reward is maximized at the point on the boundary of C_t whose tangent through that point is perpendicular to the molecule's feature vector x.

Eq. (2.6) tells that the value of the parameter R of R-sub-Gaussian noise that models some deviations of the predicted reward value against the observed reward reflects directly the size of the confidence bound. The larger the value of R is, the more the size of the confidence bound increases. In this article, the value of R is updated during the process and substituted by the upper bound of root mean square error (RMSE) on (updating) training data at each iteration step. The RMSE is calculated as follows:

$$\mathrm{RMSE}(t) = \sqrt{\frac{\sum_{t=1}^{n_t}(y_{i_t} - \hat{\theta}_t x_{i_t})^2}{n_t}}, \tag{2.9}$$

where i_t denotes the index of the molecule observed at iteration step t and $n_t = t+10$ in this article. Then the upper bound of the RMSE denoted by $\overline{R}_t$ is then defined as

$$\overline{R}_t = \sqrt{\frac{n_t}{\chi^2_{\frac{\delta}{2},n_t-1}}}\mathrm{RMSE}(t). \tag{2.10}$$

Here, $\chi^2_{\frac{\delta}{2},n_t-1}$ represents the variable that corresponds to the lower end of $[\delta/2, 1-\delta/2]$ confidence interval of chi-squared distribution with (n_t-1) degrees of freedom.

The parameter S in Eq. (2.6) is also dynamically estimated at each time step t from the l^2-norm of the upper bound of the estimated true parameter θ_* denoted by $\overline{\theta}_t$ as follows:

$$S = S_t = ||\overline{\theta}_t||_2, \tag{2.11}$$

where k^{th} element of $\overline{\theta}_t$ is expressed by

$$\overline{\theta}_{t,k} = \max\left\{\left|\hat{\theta}_{t,k} + Z\overline{R}_t\sqrt{(F_tF_t^T)_{kk}}\right|, \left|\hat{\theta}_{t,k} - Z\overline{R}_t\sqrt{(F_tF_t^T)_{kk}}\right|\right\}. \tag{2.12}$$

Here, F_t represents $(X^T_{1:n_t}X_{1:n_t} + \lambda I)^{-1}X^T_{1:n_t}$ which appears in Eq. (2.5), $(F_tF_t^T)_{kk}$ represents the k^{th} diagonal element of $(F_tF_t^T)$, and and Z corresponds to z-score, with a value of approximately ± 1.96, can be determined allowing the assumption that beyond this values $\hat{\theta}$ may exists with maximum error rate $\delta = 0.05$ The normal distribution of noise with zero mean and R^2 variance is assumed, and Eq. (2.12) gives us the upper confidence bound of estimated weight parameters under the assumptions.

There are two regularization parameters in our method. One is λ in Eq. (2.5), and the other is in LASSO for feature selection. Both parameters are tuned using grid search method, where 5-fold cross validation is adopted on training data.

Finally note that all the equations can be applied even when LASSO feature selection is adopted.

2.1.4.2 Derivation of the upper bound of expected reward

The upper bound of the expected reward is explained in Eq. (2.8), in this section we would like to derive the solution. First we recall the confidence region of $\hat{\theta}_t$ from the Eq. (2.6):

$$\begin{aligned} C_t &= \left\{ \theta \in \mathbb{R}^{N_f} : ||\theta - \hat{\theta}_t||_{\overline{V}_t} \leq \tilde{B} \right\}, \\ \text{and } \tilde{B} &= R\sqrt{2\log\left(\frac{\det(\overline{V}_t)^{1/2}\det(\lambda I)^{-1/2}}{\delta}\right)} + \lambda^{\frac{1}{2}} S. \end{aligned} \tag{2.13}$$

For each of arbitrarily chosen molecule vectors x $(= x_1, x_2, \ldots, x_{N_f})$, let $\tilde{\theta} = \underset{\theta \in C_t}{\arg\max}\ \theta^T x$ (i.e., $\tilde{\theta}$ is dependent on x). To illustrate the relationship among $\hat{\theta}_t, \tilde{\theta}$, and C_t, let us suppose a 2D case ($N_f = 2$) for simplicity (See Fig. 2.3). Because $\theta^T x$ represents the inner-product between the vectors θ and x, geometrically $\frac{\theta^T x}{|x|}$ corresponds to the distance between the origin and a point P along the vector x which perpendicularly connects to the point (θ_1, θ_2) lying within C_t. Maximizing $\theta^T x$ with respect to $\theta \in C_t$ is equivalent to maximizing $\theta^T \frac{x}{|x|}$, which makes $\tilde{\theta}$ locate on the boundary of C_t (corresponding the point Q on x). Therefore, $\tilde{\theta}$ satisfies:

$$\tilde{\theta} = \underset{\theta \in C_t}{\arg\max}\ \theta^T x \quad \text{s.t. } ||\theta - \hat{\theta}_t||_{\overline{V}_t} = \tilde{B}.$$

Let us introduce $p = \theta - \hat{\theta}_t$ for the sake of brevity. Then, we define a Lagrange function $L(p)$ with a Lagrangian multiplier $\lambda_l/2$ as follows:

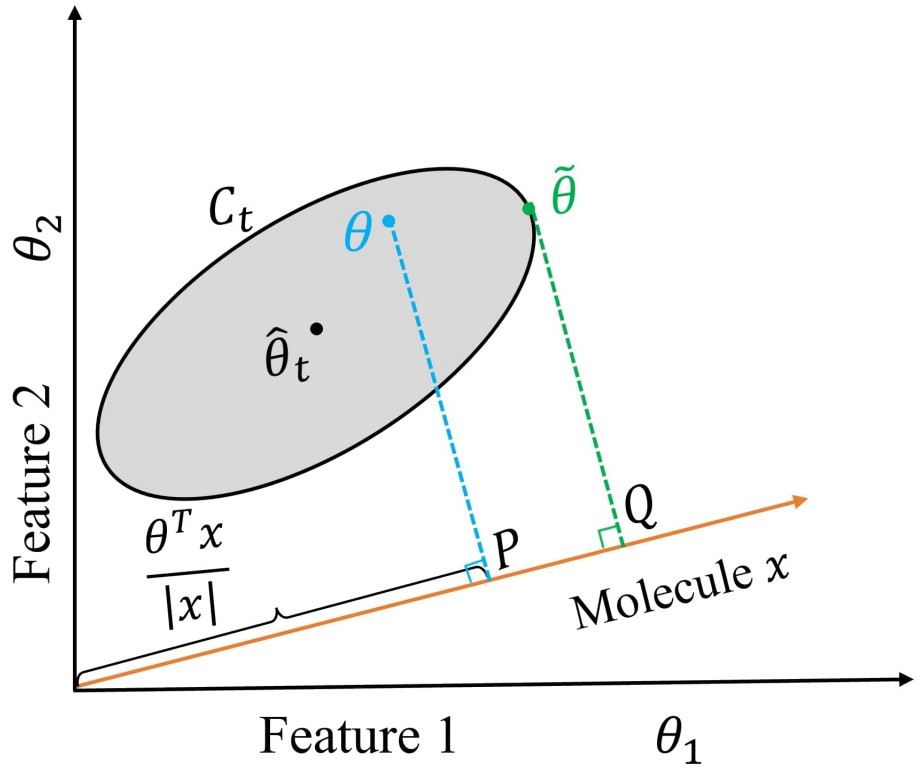

Fig. 2.3: An illustration of $\tilde{\theta}$ being located on the boundary of the confidence region C_t in the 2D case.

$$\begin{aligned} L(p) &:= \theta^T x - \frac{\lambda_l}{2}(p^T \overline{V}_t p - \tilde{B}^2) \\ &= (p + \hat{\theta}_t)^T x - \frac{\lambda_l}{2}(p^T \overline{V}_t p - \tilde{B}^2). \end{aligned} \tag{2.14}$$

Taking the differentiation with respect to p and equating the resultant $dL(p)/dp$ to zero result in the solution of p as

$$p = \frac{\overline{V}_t^{-1}}{\lambda_l} x. \tag{2.15}$$

Inserting the p into the constraint $p^T \overline{V}_t p = \tilde{B}^2$ we have

$$\frac{x^T \overline{V}_t^{-1} x}{\lambda_l^2} = \tilde{B}^2 \Leftrightarrow \lambda_l = \frac{\sqrt{x^T \overline{V}_t^{-1} x}}{\tilde{B}}. \tag{2.16}$$

Here we used the symmetric property of the matrix $\overline{V}_t^{-1}$, i.e., $(\overline{V}_t^{-1})^T = \overline{V}_t^{-1}$. From the definition of p and Eqs. (2.15) and (2.16), we get the explicit form of $\tilde{\theta}$ as follows:

$$\tilde{\theta} = \hat{\theta}_t + \frac{\tilde{B}}{\sqrt{x^T \overline{V}_t^{-1} x}} \overline{V}_t^{-1} x, \tag{2.17}$$

which implies the solution of the upper confidence bound of expected reward as follows:

$$\begin{aligned} \max_{\theta \in C_t} \theta^T x = \tilde{\theta}^T x &= \hat{\theta}_t^T x + \tilde{B} \sqrt{x^T \overline{V}_t^{-1} x} \\ &= \hat{\theta}_t^T x + \tilde{B} ||x||_{\overline{V}_t^{-1}} \end{aligned} \tag{2.18}$$

$$(\because (\overline{V}_t^{-1})^T = \overline{V}_t^{-1}).$$

2.1.4.3 Derivation of the upper bound of the root mean square error (RMSE)

In order to calculate the upper bound of RMSE, let us remind the regression model in Eq. (2.4)

$$y_i = \theta_*^T f(x_i) + \eta_i \text{ and } \eta_i \sim N(0,\ R^2), \tag{2.19}$$

where x_i denotes the vector representing a set of the molecular features associated with the i^{th} molecule, that is, $x_i := \{x_{i,k}\}$ $(i = 1, \ldots, N_{\text{tot}};\ k = 1, \ldots, N_f)$ and

η_i characterizes some deviation of the regression model from the actual reward, modelled by R-sub-Gaussian.

By the definition of χ^2-variate, we assume, at each iteration step t, the quantity $\sum_{i=1}^{n_t} \eta_i^2/R^2$ follows χ^2 distribution with $(n_t - 1)$ degrees of freedom, i.e.,

$$\frac{\sum_{i=1}^{n_t} \eta_i^2}{R^2} \sim \chi^2_{n_t-1}, \tag{2.20}$$

where $n_t = t + 10$ represents the number of the observed molecules up to time t.

At each time step the mean square error is computed by the following equation:

$$\mathrm{MSE}_{\mathrm{true}}(t) = \frac{\sum_{i=1}^{n_t} \eta_i^2}{n_t} = \frac{\sum_{t=1}^{n_t} (y_{i_t} - \theta_* x_{i_t})^2}{n_t}. \tag{2.21}$$

By combining Eqs. (2.20) and (2.21),

$$\frac{n_t \mathrm{MSE}_{\mathrm{true}}(t)}{R^2} \sim \chi^2_{n_t-1}.$$

Thus $(1-\delta)$ confidence interval for any error rate δ is defined by the following inequality:

$$P\left(\chi^2_{1-\frac{\delta}{2},n_t-1} \leq \frac{n_t \mathrm{MSE}_{\mathrm{true}}(\mathrm{t})}{R^2} \leq \chi^2_{\frac{\delta}{2},n_t-1}\right) = 1-\delta,$$

where $\chi^2_{1-\frac{\delta}{2},n_t-1}$ and $\chi^2_{\frac{\delta}{2},n_t-1}$ represent the critical values (the left and right tails, respectively) of χ^2 distribution with $(n_t - 1)$ degrees of freedom to make the probability that $\frac{n_t \mathrm{MSE}_{\mathrm{true}}(t)}{R^2}$ falls within these two values be $(1-\delta)$.

Expanding the inequality mentioned above results in

$$P\left(\frac{n_t \mathrm{MSE}_{\mathrm{true}}(t)}{\chi^2_{\frac{\delta}{2},n_t-1}} \leq R^2 \leq \frac{n_t \mathrm{MSE}_{\mathrm{true}}(t)}{\chi^2_{1-\frac{\delta}{2},n_t-1}}\right) = 1-\delta.$$

By using $\mathrm{RMSE}_{\mathrm{true}}(t) = \sqrt{\mathrm{MSE}_{\mathrm{true}}(t)}$ and taking a square root on both sides, we can rewrite the above relation to

$$P\left(\sqrt{\frac{n_t}{\chi^2_{\frac{\delta}{2},n_t-1}}}\mathrm{RMSE}_{\mathrm{true}}(t) \leq R \leq \sqrt{\frac{n_t}{\chi^2_{1-\frac{\delta}{2},n_t-1}}}\mathrm{RMSE}_{\mathrm{true}}(t)\right) = 1-\delta. \tag{2.22}$$

This implies the upper bound of RMSE, denoted by $\overline{R}_t$, is given by

$$\overline{R}_t = \sqrt{\frac{n_t}{\chi^2_{1-\frac{\delta}{2},n_t-1}}}\mathrm{RMSE}_{\mathrm{true}}(t). \tag{2.23}$$

Since the true $\mathrm{RMSE}_{\mathrm{true}}(t)$ is unknown in general, we use an estimated value $\mathrm{RMSE}(t) = \sqrt{\frac{\sum_{i=1}^{n_t}(y_{it}-\hat{\theta}_t x_{it})^2}{n_t}}$ for $\mathrm{RMSE}_{\mathrm{true}}(t)$.

2.1.4.4 Derivation of the upper bound of the true regression parameter

We would like to calculate the upper bound of the true regression parameter θ_*. At each iteration step t, $\hat{\theta}_t$ is computed by Eq. (2.5):

$$\hat{\theta}_t = (X^T_{1:n_t}X_{1:n_t} + \lambda I)^{-1}X^T_{1:n_t}y_{1:n_t}.$$

The variance-covariance matrix of $\hat{\theta}_t$ can be calculated using variance operator as follows:[52]

$$\mathrm{Var}(\hat{\theta}_t) = \mathrm{Var}\left((X^T_{1:n_t}X_{1:n_t} + \lambda I)^{-1}X^T_{1:n_t}y_{1:n_t}\right)$$

Now we consider $F_t := (X^T_{1:n_t}X_{1:n_t} + \lambda I)^{-1}X^T_{1:n_t}$ is a constant matrix of $N_f \times n_t$ and the variance-covariance matrix of $y_{1:n_t}$ is $R^2 I$. Using $\mathrm{Var}(Cx) = C\mathrm{Var}(x)C^T$

where C and x are a constant matrix of $n \times m$ and an m-dimensional vector (n, m : any positive integer), respectively, the variance-covariance matrix of $\hat{\theta}_t$ is derived as:

$$\begin{aligned} \mathrm{Var}(\hat{\theta}_t) &= F_t \mathrm{Var}(y_{1:n_t}) F_t^T \\ &= (F_t F_t^T) R^2. \end{aligned} \tag{2.24}$$

Hence, $(F_t^T F_t)_{kk} R^2$ is the variance of $\hat{\theta}_{t,k}$. A confidence interval for $\theta_{*,k}$ is evaluated by

$$\left[\hat{\theta}_{t,k} - ZR\sqrt{(F_t F_t^T)_{kk}},\ \ \hat{\theta}_{t,k} + ZR\sqrt{(F_t F_t^T)_{kk}} \right] \tag{2.25}$$

where $R\sqrt{(F_t F_t^T)_{kk}}$ corresponds to the standard deviation of $\hat{\theta}_{t,k}$ and Z corresponds to a parameter to meet the allowance error rate δ,

in particular $Z = 1.96$ for $\delta = 0.05$. However, since R in the above expression is unknown, we use $\overline{R}_t$ from Eq. (2.23). Hence the confidence interval of $\theta_{*,k}$ is computed as:

$$\left[\hat{\theta}_{t,k} - Z\overline{R}_t\sqrt{(F_t F_t^T)_{kk}},\ \ \hat{\theta}_{t,k} + Z\overline{R}_t\sqrt{(F_t F_t^T)_{kk}} \right] \tag{2.26}$$

2.2 Fixed budget and fixed confidence settings

Fixed budget and fixed confidence settings are typically associated with the exploration-exploitation dilemma in multi-armed bandit problems, where an

agent needs to decide which actions (arms) to choose to maximize its cumulative reward over time. Here, I'll explain a couple of algorithms related to each setting:

Fixed Budget Bandit Algorithm: In the fixed budget scenario, the agent has a predetermined number of trials or budget. The algorithm aims to balance exploration and exploitation by choosing arms within this predetermined number of trials or budget. For example UCB1 algorithm selects arms that have both high estimated rewards and high uncertainty (captured by the confidence bounds) to explore different options efficiently within the given budget.

Fixed Confidence Bandit Algorithm: In this setting, the agent continues to sample arms until it is confident that it has identified the arm with the highest true reward. The algorithm progressively eliminates sub-optimal arms as it gains more confidence in the arm with the highest reward. The choice between fixed budget and fixed confidence depends on the application and the available resources or constraints in terms of the number of trials or the required confidence level. Fixed Confidence Bandit Algorithm have been used in this thesis.

2.3 Methods for comparison

The OFUL bandit algorithm is compared with traditional methods, such as Bayesian optimization (BO) and Random Search (RS), to observe the performance.

2.3.1 Bayesian optimization

Bayesian optimization uses a Gaussian process regression as a surrogate function with a radial basis function (RBF) kernel (Other kernel for example Tan-

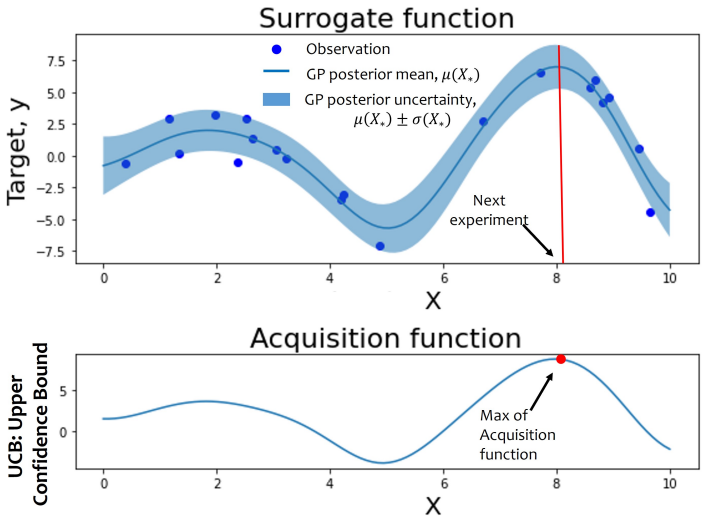

Fig. 2.4: Graphical presentation Bayesian Optimization (BO). Upper figure corresponds to Gaussian Process(GP) surrogate function and Lower figure represents the acquisition function. This research uses upper confidence bound acquisition function.

imoto kernel, Dot product kernel or Matern kernel also possible. Graphical presentation of the Bayesian Optimization (BO) is refereed in Fig.2.4. The candidate $x_{i_{t+1}}$ is recommended at iteration step t by the acquisition function studied by Gotovos et al.[53,54] defined as follows:

$$x_{i_{t+1}} = \underset{x \in D_{t+1}}{\arg\max} \left(\hat{y}(x) + \beta_t^{\frac{1}{2}} \sigma(x) \right), \tag{2.27}$$

where $\hat{y}(x)$ and $\sigma(x)$ are the estimated expected value and the standard deviation for the target value of molecule whose feature vector is represented by x in the unobserved set D_{t+1}, respectively, and $\beta_t = 2\log\left(\frac{|D_{t+1}|\pi^2 t^2}{6\delta}\right)$ where δ (typically used 0.05) corresponds to the allowance error rate, as applied in the context of OFUL bandit algorithms and $|D_{t+1}|$ denotes the number of the unobserved molecules remaining at time t in the pool of candidates. A data-driven approach was employed to determine the searching range of the length scale hyperparameter used in the RBF kernel. The range was chosen as the minimum to the maximum of the Euclidean distances between the observed data x_i up to

each iteration time. The upper figure shows the Gaussian Process (GP) surrogate function, while the lower figure displays the acquisition function. This research employs the upper confidence bound acquisition function. The red points in the acquisition function indicate the maxima, and the corresponding molecule will be examined in the next experiment, as indicated by the red vertical line in the surrogate function.

2.3.2 Random search

In addition to OFUL and BO, a random search (RS) policy for selecting the sequence of candidates was also implemented. While this policy randomly selects each candidate throughout the process from the entire database of molecules in a uniform manner, the stopping condition for RS is defined using Eq. (2.4) and Eq. (2.5), similar to OFUL.

2.4 Stopping condition

The construction of stopping condition is crucial to terminate the method automatically. The algorithm stops testing all the N_{tot} candidates or satisfying the following conditions. Obviously when all the candidates are tested, the best candidate for achieving the maximum reward could be identified. However, it should be desired to identify the best one more efficiently with fewer number of experiments with ensuring accuracy more than $1-\delta$ within the framework of Eq. (2.4). The stopping condition for OFUL at iteration step t is given by

$$y_{\max}(t) > \max_{(x,\theta)\in D_{t+1}\times C_t} \theta^T x, \tag{2.28}$$

where $y_{\max}(t)$ denotes the maximum value among the observed rewards up to iteration step t, and the right-hand side of equation expresses the upper confidence bound of expected reward at iteration step $t+1$ within the confidence region of weight parameter constructed at iteration step t. This means that the iteration should be stopped at time t when there are no potential candidates that might become larger than the present maximum. Note that random search (RS) method also uses this stopping condition because it shares the same prediction model of OFUL.

The stopping condition for Bayesian optimization (BO) is constructed by the following equation derived from Eq. (2.27):

$$y_{\max}(t) > \max_{x \in D_{t+1}} \left(\hat{y}(x) + \beta^{\frac{1}{2}} \sigma(x) \right), \tag{2.29}$$

where the right-hand side represents the maximum of the upper confidence bound among the unobserved data in the BO framework.

For both cases, a graphical representation of the stopping condition is shown in Fig.2.5. In this figure, green points represent observed data, while red points denote unobserved data. Since observed data has no uncertainty, only the uncertainty for unobserved data is calculated, as represented by the vertical red lines. These lines indicate the confidence interval, computed using the right-hand side of Eq. 2.28 and Eq. 2.29. If any upper bound of the confidence interval for unobserved data exceeds the horizontal blue line ($y_{\max}$) the next experiment should be conducted (Fig. 2.5(a)). Otherwise, the experiment should be stopped (Fig. 2.5(b)).

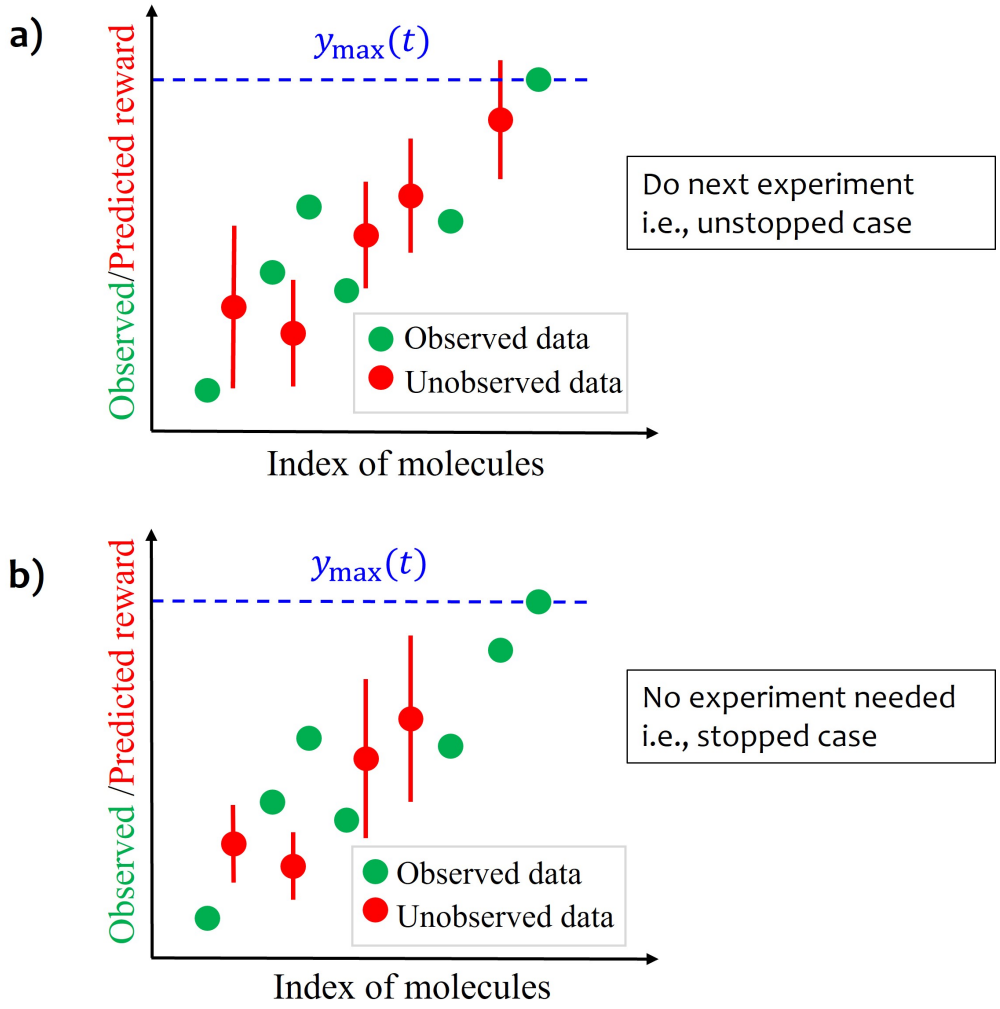

Fig. 2.5: Pictorial presentation of stopping condition. Figure (a) presents the non-stopped case because the upper confidence bound of the unobserved data exceeds the threshold and figure (b) presents the stopped case because the upper bound of the threshold which means there is no possibility to exists a molecule having molecular property larger than $y_{max}(t)$

2.5 Molecular descriptors

According to Todeschini and Consonni,[55] a molecular descriptor is "the final result of a logic and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardized experiment." Molecular descriptors can be categorized into two types: i) experimental measurements, such as *logP*, molar refractivity, dipole moment, and polarizability, and ii) theoretical molecular descriptors, which are derived from a symbolic representation of the molecule. Theoretical molecular descriptors can be further classified based on different types of molecular representation and can be calculated using SMILES (Simplified Molecular Input Line Entry System) notation for molecules.

Various molecular descriptor tools are employed in chemical research, including alvaDesc, Dragon, ISIDA descriptors, Morgan fingerprints, Mordred descriptors, PaDEL-descriptor, RDKit, and others. Some of these tools are available for free, while others require commercial licenses. In this research, ISIDA fragment descriptors were calculated. The calculation procedure will be presented in the **Descriptor/feature computation** section.

2.5.1 SMILES code

SMILES (Simplified Molecular Input Line Entry System) is a method for representing chemical structures using ASCII (American Standard Code for Information Interchange) strings. It's a compact and human-readable way to describe molecules and reactions. SMILES strings are based on the connection table representation of molecules. In a SMILES string, atoms are represented by their atomic symbols (e.g., C for carbon, O for oxygen), and bonds between atoms

are indicated by various symbols such as "-", "=", "#", etc. For example SMILEs code for Water (H2O), Ethanol (CH3CH2OH), and Benzene (C6H6) are "O", "CCO" and "c1ccccc1" respectively.

2.5.2 Fragment features

CGRtools[56] library and its native substructure extraction functions are used to determine substructures within a molecule. The size of these substructures is defined by specified lower and upper limits, which correspond to the topological radius—the number of atoms away from a central atom (including the central atom itself). A radius of 1 includes only the atom itself, a radius of 2 includes the atom and all atoms directly connected to it, and so on. This process is repeated for all atoms in the molecule/CGR across the range of sizes from the lower to the upper limit to construct the fragment table.

ISIDA/Fragmentor-2017 allows for the creation of both linear and atom-centered fragments. Atom-centered fragments are essentially collections of linear fragments that share a central atom, without explicitly representing branching or rings. Our implementation emulates this functionality.

Therefore the total extracted numbers of features are 1775, 313 and 510 for FreeSolv and enantioselectivity and Photoswitch dataset respectively. To avoid the confusion of the feature number here we again presents the number feature after reprocessing which are 447, 148 and 398 for FreeSolv and enantioselectivity and Photoswitch dataset respectively.

2.5.3 Morgan fingerprint features

The Morgan fingerprint features were also calculated to compare with ISIDA (In Silico Design and Data Analysis) fragment features. It was observed that

Morgan fingerprints are less effective for the problem setting discussed in the **Results and Discussion** chapter. The key differences between Morgan fingerprints and ISIDA fragment descriptors are as follows: the former only indicates whether a substructure is present or absent within a defined radius (binary vector),[57] whereas the latter counts the number of substructures present within a defined radius.[47,58] Additionally, fragment descriptor features offer distinct advantages, such as the ability to uniquely represent any molecular graph invariant, express any symmetric similarity measure, and facilitate the representation of regression or classification models.[59]

2.6 Data preparation

In the extensive array of features, some are deemed irrelevant and subsequently removed from the dataset. Elimination criteria include features that are duplicated, those with a correlation coefficient of one, and features with an extremely small number of non-zero values, specifically, with a sum of entries up to five. Following this process, the total number of retained features tallies 447 for the FreeSolv dataset and 148 for the enantioselectivity dataset. Additionally, data standardization is performed at each stage of the analysis. The standardization formula can be expressed as follows: $z_i = (x_i - \bar{x})/\sigma_i$, where $\bar{x}$ and σ_i represent the mean and standard deviation of the original feature x_i, respectively. The resulting standardized feature, denoted as z_i, guarantees a mean of 0 and a variance of 1 across all features.

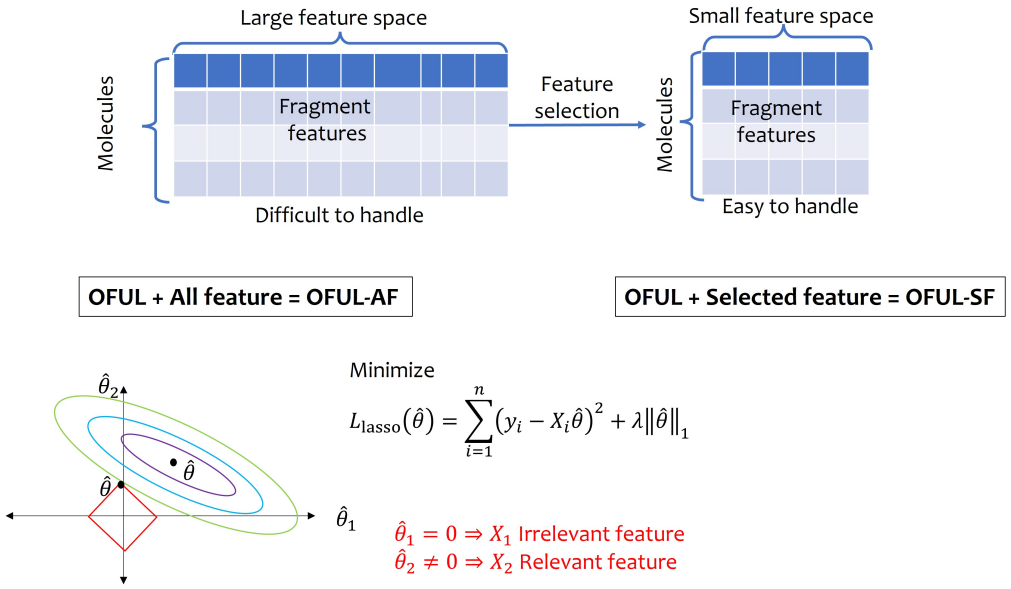

Fig. 2.6: Working strategy of LASSO regression.

2.7 Dynamic feature selection

The data typically consist of a couple of hundreds features, among them some may not be relevant to the molecular property to predict. Consequently, removing the redundant and irrelevant features is an important step for optimization task in the huge dimensional feature space. The LASSO (Least Absolute Shrinkage and Selection Operation) regression analysis by Hastie et. al.[60] was applied for this purpose. LASSO is a penalized regression analysis technique that uses l^1 regularization, which can efficiently reduce the number of features for linear regression model (see the Fig.2.6). To decide the regularization parameter of the LASSO model, we apply grid search with 5-fold cross validation for samples observed so far. The OFUL bandit algorithm was examined with two different feature sets: all features (AF), which remain invariant throughout the process, and selected features (SF), which are updated dynamically. These are referred to as OFUL-AF and OFUL-SF, respectively. For comparison, two

versions of Bayesian Optimization (BO) were also examined, labeled as BO-AF and BO-SF. It is important to note that the Random Search (RS) approach does not involve any feature selection.

Because each feature has different range of the values with different physical units. Thus, standardization method have been employed i.e., subtracts the mean ($\bar{x}_k$) averaged over all molecules in the pool of the candidates from the k^{th} feature variable of the i^{th} molecule ($x_{i,k}$) with division by the standard deviation (σ_k), resulting in a standardized feature variable, i.e., $x_{i,k} \leftarrow (x_{i,k} - \bar{x}_k)/\sigma_k$.

2.8 Conclusion

This chapter discusses the concept of the sequential optimization method in chemistry. The Bandit algorithm, specifically the Optimism in the Face of Uncertainty Linear (OFUL) bandit, is covered along with Bayesian Optimization (BO). Additionally, the implementation of the dynamic feature selection method and stopping condition are detailed. Finally, the process of calculating the features is explained.

Chapter 3

Sequential Optimization of Synthetic Data with Different Ratio of Relevant Features

This chapter is designed to introduce the data we utilized in this research endeavor. The aim of this thesis work was to develop a new algorithm for sequential optimization of molecular property. We develop a modified version of Optimism in the Face of Uncertainty Linear (OFUL) bandit along with automatic stopping condition which enables to stop algorithm after getting the desired molecule immediately. In this regard we consider two real data set and two synthetic data set to validate our developed algorithm. The inclusion of both synthetic and real data aims to demonstrate the method's generalizability.

3.1 Synthetic data

Two synthetic datasets, named Syn-I and Syn-II, are constructed using a linear regression model with Gaussian noise. We define a set of N_{tot} candidates

$y = \{y_i\}$ approximately characterized by a set of N_f feature variables $x = \{x_{i,k}\}$, expressed as $y_i = \sum_{k=1}^{N_f} \theta_k x_{i,k} + \eta_i$, where $i = 1, ..., N_{tot}$ and $k = 1, ..., N_f$. We set a problem to find the candidate i_m whose y_{i_m} value is the largest as fewer number of experiments as possible. The feature variables intrinsic to each candidate are drawn from Gaussian distribution with mean 0 and variance 1, $N(0,1)$. In both datasets, the weight parameter θ_k and Gaussian noise η_i are also sampled from $N(0,1)$. The first dataset, Syn-I, and the second dataset, Syn-II, consist of $N = 300$ candidates and $N_f = 30$ feature variables, and $N = 300$ candidates and $N_f = 500$ feature variables, respectively. In the former, among the 30 feature variables, randomly chosen 10 are considered to be relevant and important (i.e., the corresponding weight parameters are nonzero otherwise zero). In the latter Syn-II only 30 out of 500 are assumed to be relevant. We examine two synthetic datasets to tackle the issue of higher dimensionality. Syn-I dataset is generated with fewer features than sample points (akin to real-world data molecules), making it less complex. Conversely, Syn-II presents a greater challenge with a larger number of features compared to sample points. In real application we don't know whether all feature are relevant or not, in fact most of the time some irrelevant feature exists thus feature selection method needed to cut off the features. Therefore, the concept of relevant and irrelevant feature is introduced to acknowledge necessity of feature selection methods as well as to diminish the burden of huge feature.

3.2 Brief discussion of methods

Before going to deep into the results of synthetic dataset, here, protocol of the linear bandit algorithm for the selection of potential candidate molecules

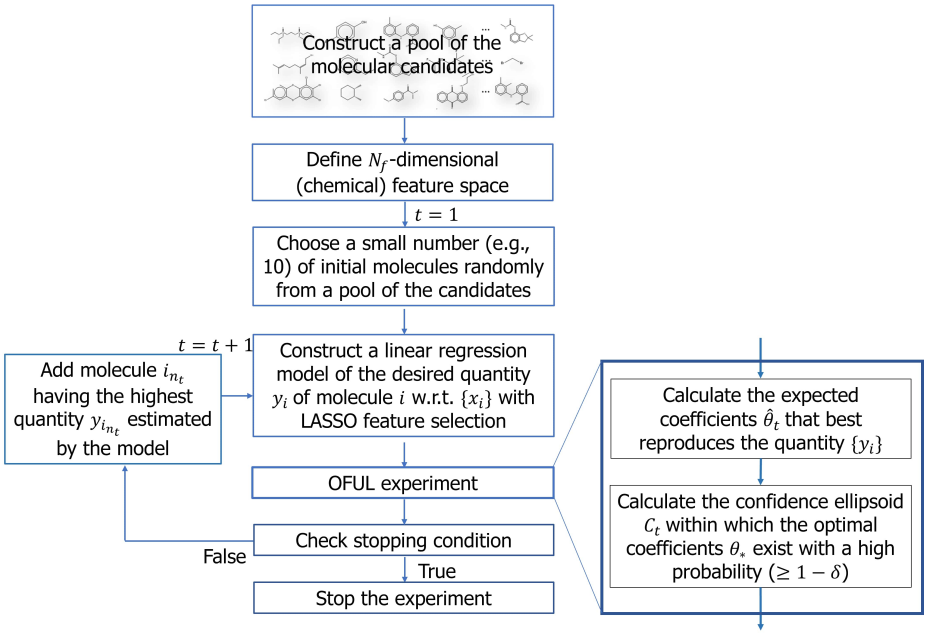

Fig. 3.1: An overview of our OFUL bandit algorithm combined with dynamic feature selection.

is briefly summarized, with the aim of optimizing specific molecular properties (see also Fig. 3.1). Prior to beginning of the algorithm, a set of potential candidates (=molecules) is prepared in advance. Additionally, molecular features for the set of candidates are extracted using standard chemo-informatics tools.[56,61,47] If a molecule in the primary set of the candidates has only less than or equal to five nonzero feature values, such a molecule are removed from the consequent analysis, and all feature variables are standardized as mean zero and variance one.

The standardization method involves subtracting the mean ($\bar{x}_k$) averaged over all molecules from the k^{th} feature variable of the i^{th} molecule ($x_{i,k}$) and dividing by the standard deviation (σ_k), resulting in a standardized feature variable denoted by $z_{i,k}$, i.e., $z_{i,k} = (x_{i,k} - \bar{x}_k)/\sigma_k$. The algorithm begins with randomly selecting an initial set of molecules (say, 10 molecules) from the pool of candidates and includes them in the training data set. Note here that we only

utilize the molecular property (reward in MAB algorithm) associated with the training data set and suppose that the actual values of the target molecular property of the remaining candidates in the pool are unknown (just similarly to real experiments). With removing these initial set of molecules from the candidate pool, the first step of the algorithm is to construct a linear regression model from the chosen training data set of molecules, to predict the molecular property for the remaining data in the pool. Here, compared to the total number of molecular descriptors and the number of molecules to be examined, the former is usually much larger than the latter. Thus, LASSO feature selection[60] is applied to the (initial) set of molecular features, and the selected features are used for modelling. Next, the algorithm suggests a new molecule from the remaining candidates based on the expected reward and its uncertainty corresponding to its approximated confidence bound, evaluated using optimism in the face of uncertainty (OFU) strategy[31] (See the details in **Chapter 2: Theoretical Developments on Sequential Optimization**). But if LASSO doesn't find good feature(s), algorithm suggests a molecule randomly. The molecule that is predicted to have a larger value of the molecular property than the largest "observed" among the training set is then "tested". In this study, "testing" corresponds to obtaining the result of the molecular property from the accumulated benchmark set, as equivalent to a real experiment. With removing the test molecule from the candidate pool, these processes are repeated until a condition is met by using the uncertainty of predicted rewards in the model to guarantee with a large probability that the largest reward obtained insofar is also the largest in all candidates involving those that has not been examined (the condition is termed as stopping condition, and the time round satisfying the stopping condition is called stopping time). In this research, if the stopping condition is met and the largest

reward is actually found (in these simulation experiments, the ground truth is known whilst in real experiments we do not), such run is considered as success; otherwise, failure.

In this research, bandit algorithm is evaluated by using four benchmark datasets, two synthetic data and two real datasets. Two synthetic datasets, named Syn-I and Syn-II, are constructed using a linear regression model with Gaussian noise. We define a set of candidates $y = \{y_i\}, (i = 1, ..., N)$ or $y = \{y_i \mid i = 1, ..., N\}$ approximately characterized by a set of feature variables $x = \{x_{i,k}\}, (k = 1, ..., N_f)$, expressed as $y_i = \sum_{k=1}^{N_f} \theta_k x_{i,k} + \eta_i$, and set a problem to find the candidate i_m whose y_{i_m} value is the largest as fewer number of experiments as possible. The feature variables intrinsic to each candidate are drawn from Gaussian distribution with mean 0 and variance 1, $N(0,1)$. In both datasets, the weight parameter θ_k and Gaussian noise η_i are also sampled from $N(0,1)$. The first dataset, Syn-I, and the second dataset, Syn-II, consist of $N = 300$ candidates and $N_f = 30$ feature variables, and $N = 300$ candidates and $N_f = 500$ feature variables, respectively. In the former, among the 30 feature variables, randomly chosen 10 are considered to be relevant and important (i.e., the corresponding weight parameters are nonzero otherwise zero). In the latter Syn-II only 30 out of 500 are assumed to be relevant. The real datasets include experimental hydration free energies of 642 molecules (FreeSolv dataset),[46] experimental free energy difference between enantiomer products ($\Delta\Delta G$) of 70 reaction systems (enantioselectivity dataset)[47] and wavelength of 392 molecules (Photoswitch dataset).[48,49] The problem is set to find the lowest hydration free energy for FreeSolv data, the maximum absolute values of $\Delta\Delta G$ for enantioselectivity data and highest wavelength for Photoswitch data. The total numbers of features are 447, 148 and 398 respectively for FreeSolv, enantioselectivity and Photoswitch

dataset, which are fragment descriptors based on SMILES (Simplified Molecular Input Line Entry System) codes of molecules.[47]

To validate the performance of our algorithm, we implement five algorithms: OFUL-AF and OFUL-SF, which skips and adopts feature selection, respectively, Bayesian optimization without feature selection (BO-AF), with feature selection (BO-SF) and random search (RS) which randomly suggests next candidate. We have repeated the simulation several times (50 times) and each of the repetitions is termed as run. In the case of synthetic data analysis, every run uses different data of the same size which are generated randomly as explained before in this section. The results are produced using typical error rate, $\delta = 0.05$ and the statistics are based on 50 independent runs.

3.3 Sequential optimization of Synthetic data

Figure 3.2 (a) presents box plots of the time steps required to find the optimal candidate (ground truth), which we called the best finding time, and the stopping time (=the time satisfying the stopping condition) for Syn-I. The best finding time can be computed if and only if the ground truth is known a priori (of course this is not the case in real applications). This quantity is only referred in addressing each algorithm performance. The orange lines and green triangles indicate the median and mean, respectively. For the best finding time, the median values are 135.0, 11.0, 6.0, 13.0, and 10.0 iteration steps for RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, respectively. The results show that RS is significantly slower than the other four methods in finding the best candidate, which confirms the importance of model-based sequential optimizations.

Moreover, the variance of best finding time for RS is much larger than that

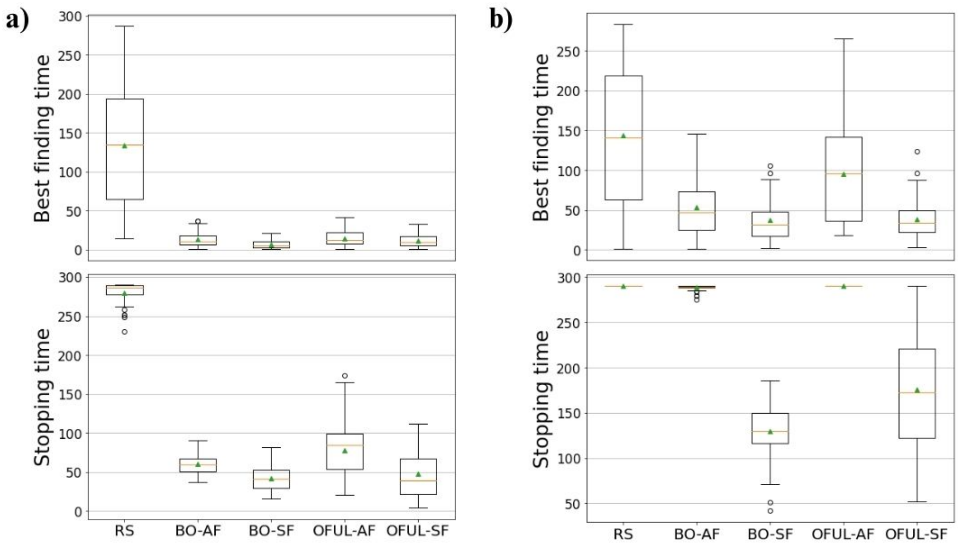

Fig. 3.2: Box plots of best finding time (upper) and stopping time (lower) for (a) Syn-I and (b) Syn-II. The total numbers of elements in Syn-I and Syn-II data set are 300 both, and the experiments are initiated with 10 randomly chosen seeds. Syn-I has 10 relevant feature variables out of 30 variables, and Syn-II does 30 out of 500.

of the other four methods. Regarding the stopping time, the median values are 286.0, 59.0, 41.0, 85.0, and 39.0 iteration steps for RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, respectively. Although RS cannot judge to stop in most cases within the maximum number of iterations "290 (=300 candidates - 10 randomly chosen initial seeds)", the other four methods were successful in finding the best candidate and automatically stopping at an early stage of iterations, i.e., the success rates of BO-AF, BO-SF, OFUL-AF, and OFUL-SF are 100%, 100%, 94%, and 98%, respectively. Also, it is observed that BO and OFUL improve by incorporating dynamic feature selection and the OFUL-SF and BO-SF show comparable performance both in best finding time and stopping time. Here, the existence of failure cases by any algorithm (e.g., 2% by OFUL-SF) means that it could not find the underlying ground truth but before reaching the maximum number of iterations the algorithm stopped with satisfying the stopping condition. Note in this case that one cannot define the best finding time, i.e., the time

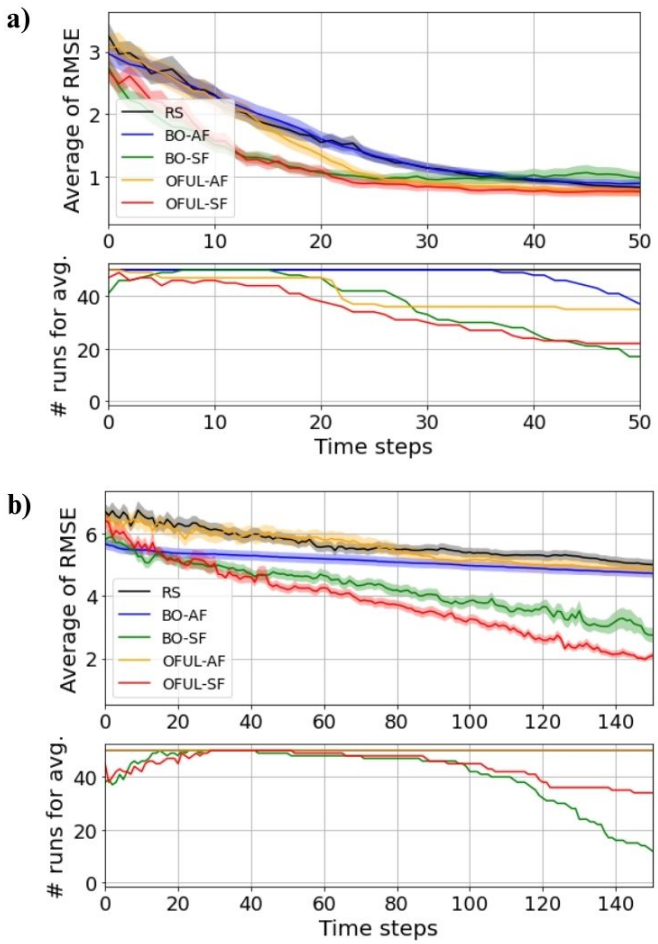

Fig. 3.3: RMSE analysis for unobserved candidates at each time step, (a) Syn-I and (b) Syn-II. 50 different runs with different initial seeds, and each colored shaded area corresponds to $\pm\sigma$ of the statistics. The number of runs used for statistics is also shown in the lower part. The RS, OFUL-AF, and BO-AF lines for Syn-II are overlapped completely until time step 150.

to find the ground truth in the iteration process, and hence such cases were not included in the figure (e.g., the upper figure in Fig. 3.2(a))

Figure 3.2 (b) presents boxplots of the best finding time and the stopping time for Syn-II. The median values of best finding time are 140.5, 47.0, 31.5, 96.5, and 35.5 iteration steps for RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, respectively. The values are larger than those in Syn-I, which indicates the difficulty in Syn-II where many irrelevant feature variables are included. It is

found that dynamic feature selection can improve the best finding time both in BO and OFUL, and the variance of best finding time is also reduced. For the stopping time, median values are 290.0, 289.5, 130.0, 290.0, and 172.5 iteration steps for RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, respectively. Notably RS, BO-AF, and OFUL-AF reached the maximum of iteration steps "290", and only BO-SF and OFUL-SF were successful in finding the best candidate and stopping within the limit, i.e., the success rates of BO-SF and OFUL-SF are both 100%. The results show that dynamic feature selection is essential in Syn-II, to which many realistic chemical problems are expected to belong, as in the enantioselectivity dataset below. Although the best finding times of BO-SF and OFUL-SF are comparable, the stopping times of BO-SF tend to be faster in this case. It should be noted in the comparison between Syn-I and Syn-II that the effect of dynamic feature selection is much more significant in Syn-II than in Syn-I.

Figure 3.3 (a) and (b) present the accuracy of each model of RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, updated during the sequential optimization, for Syn-I and Syn-II data, respectively. The accuracy is evaluated by using the root-mean-square error (RMSE) for the difference between the predicted values of unobserved candidates at each time step and their true values. The averages and standard errors of RMSE at each time step are calculated for the valid runs that are unstopped yet and at least one feature is selected by feature selection at that time step. Also, note that RS utilized the same regression model and stopping condition as OFUL except the candidate selection policy. It is found that RMSEs of all the methods decay gradually as the time step increases, and especially the decays of OFUL-SF and BO-SF are faster than those of the other methods in both datasets. The decreases in RMSEs of these two methods with

feature selection are more pronounced at longer iteration steps for the Syn-II case, as being consistent with the larger effect of dynamic feature selection on best finding and stopping times in Syn-II rather than in Syn-I. The comparison of RMSEs between BO-SF and OFUL-SF tells that OFUL-SF predicts the unobserved candidates y more accurately than BO-SF does (on average) in both Syn-I and Syn-II. This makes us expect that OFUL-SF should exhibit better performance in the best finding and stopping time analysis. However, the distinction between the performance of BO and OFUL methods in best finding and stopping times is nontrivial, attributed to the different construction of confidence intervals or uncertainties in the predicted values. Within the OFUL framework, the reduction of feature variables in regression is crucial for reducing the confidence region to stop properly as known from the Eq. (2.6) in the Methods and Materials section. Similarly, in the BO framework, the variance assessed through Gaussian process regression is employed, with the variances significantly reduced by feature selection. These factors are found to be essential in comprehending the substantial differences in stopping times between the methods with and without dynamic feature selection, observed in the Syn-II results. The different construction of confidence intervals is also related to the observation that the stopping times of BO-SF are faster than those of OFUL-SF in the Syn-II, despite the RMSE of BO-SF being larger than that of OFUL-SF at iteration steps beyond $\sim$40. Additionally, the stopping times of RS in Syn-I are slightly smaller than those in Syn-II, where the confidence intervals are influenced by the difference in the total number of features.

3.4 Conclusion

In this chapter we explain the synthetic data generation and application of the newly developed algorithm. We observed that OFUL bandit algorithm is comparable with traditionally used Bayesian optimization method.

Chapter 4

Sequential Optimization of Experimental Data of Hydration Free Energies

4.1 Brief discussion of methods

The linear bandit algorithm (OFUL bandit algorithm) and Bayesian optimization (BO), along with their stopping conditions, are discussed in **Chapter 2**. A brief overview of these methods is also provided in **Chapter 3**. To save time and space, the same discussion has been excluded from this chapter.

4.2 Free solvation energy (FreeSolv) data

An algorithm was developed to identify the best molecules while minimizing the number of experiments required. To establish the algorithm, three real datasets and two synthetic datasets were considered. The synthetic data are explained in **Chapter 3**. This chapter focuses on the real data. The real datasets include: i)

The solubility of molecules in water, measured as hydration free energies ΔG^{hyd} of 642 molecules, known as the FreeSolv dataset,[46] frequently used in chemical synthesis, sequential optimization, and drug screening research. This data set will be discussed in the current chapter. ii) The free energy barrier difference between enantiomer products ($\Delta\Delta G$) of 70 reaction systems, extracted by Tsuji et al.,[47] referred to as the enantioselectivity dataset. THis data set will be discussed **chapter 5**, and iii) The photophysical properties of molecules, specifically the $\pi - \pi^*$ transition wavelength of the E isomer, known as the Photoswitch dataset,[48,49] consisting of 392 molecules. This dataset is used as a benchmark for wavelength prediction and multi-objective optimization, although only a single target is considered in this research due to the single-objective optimization setting. Photoswitch dataset will be discussed in **Chapter 6**.

For the FreeSolv data, the problem is defined as finding the molecule with the lowest hydration free energy. For the enantioselectivity data, the goal is to find the maximum absolute value of the energy barrier difference between enantiomer products, i.e., $\max|\Delta\Delta G|$. For the photoswitch data, the objective is to determine the maximum wavelength using as few computations (experiments) as possible.

Among the three real data this chapter discuss only FreeSolv data (subsequent chapter will be employed for the other two datasets). Details of the FreeSolv dataset is explained now on. The term Hydration free Energy (ΔG^{hyd}) is a specific case of Solvation free energy (ΔG^{solv}). Here let's define Solvation free energy (ΔG^{solv}) first. It is nothing but the free energy change associated with the process of dissolving a solute in a solvent, regardless of the type of solvent involved and on the other hand hydration free Energy (ΔG^{hyd}) refers to the free energy change associated with the process of dissolving a solute in water.

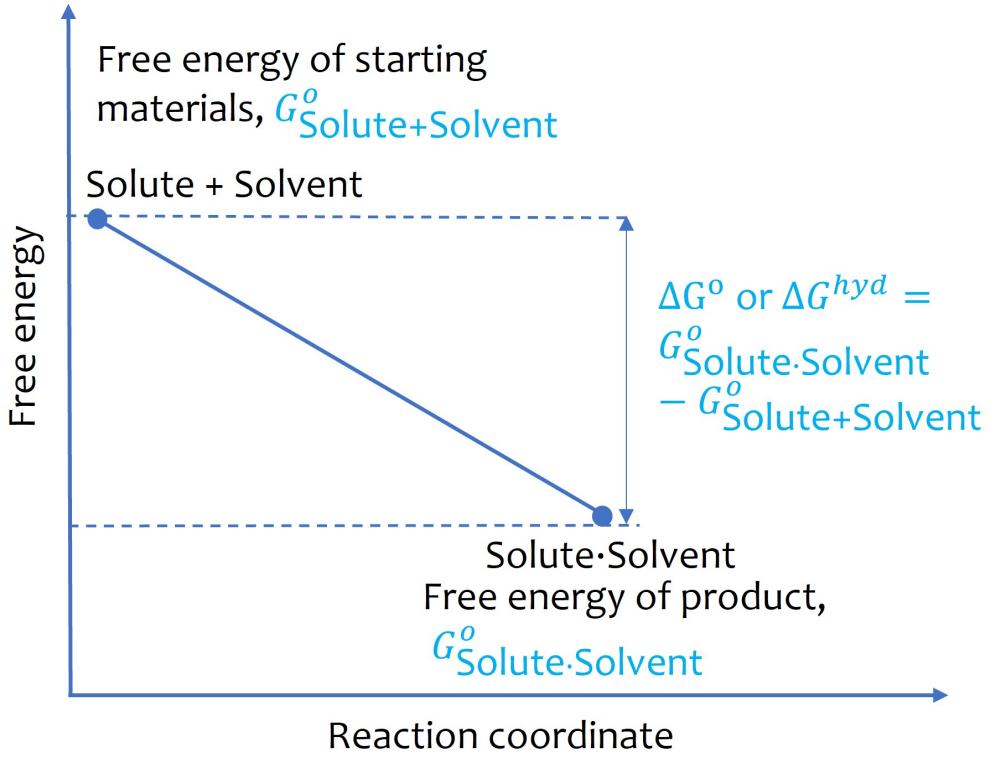

Fig. 4.1: Definition of Hydration free Energy (ΔG^{hyd}) using reaction coordinate diagram.

To explain the definition of hydration free energy (ΔG^{hyd}) reaction coordinate diagram have been drawn(see Fig. 4.1), which could make reader easy to understand the phenomena. Here X and Y axes represents the "Reaction coordinate" and "Free energy" respectively.

The point indicated by "Solute + Solvent" is the initial state of the process, where "+" sign refers that both materials are separated i.e. still they have not form solution. The horizontal dotted line is the free energy of the "Solute + Solvent". Now the free energy of the initial stage is $G^o_{Solute+Solvent}$.

Essentially, Energy barrier or activation energy is the minimum energy required for the reactant to transform into the product. Then it make the solution denoted by "Solute · Solvent" here (·) between the "Solute" and "Solvent" refers

that materials are not separated i.e. solution formed. Now the right foothill of the figure shows that the free energy of the solution is $G^{o}_{Solute \cdot Solvent}$ thus the hydration free energy (ΔG^{hyd}) is define by the difference between free energy of initial materials $G^{o}_{Solute+Solvent}$ and free energy of solution $G^{o}_{Solute \cdot Solvent}$ more explicitly the formula can be written as $\Delta G^{hyd} = G^{o}_{Solute \cdot Solvent} - G^{o}_{Solute+Solvent}$. The goal is to find the molecule with the lowest hydration free energy ΔG^{hyd} within as few experiments as possible.

4.3 Free solvation energy data analysis

Figure 4.2 presents boxplots of the best finding time and the stopping time for FreeSolv data, where total number of molecules is 642 with total number of features is 447. The median values for the best finding time are 370.0, 7.5, 11.0, 7.0, and 19.0 iteration steps for RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, respectively. As for the stopping time, the median values are 632.0, 97.0, 34.0, 499.5, and 68.0 iteration steps for RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, respectively. Note that the observed trends in best finding times and stopping times align notably with those of Syn-I rather than Syn-II, except for the OFUL-AF result. The discernible dissimilarity between BO-AF and OFUL-AF in stopping times is more pronounced compared to synthetic datasets, with OFUL-AF exhibiting difficulty in achieving a reduction in confidence intervals due to a large number of total feature variables in FreeSolv case, akin to the Syn-II scenario. In contrast, differences in variances between BO-AF and BO-SF are smaller than those in Syn-II case and akin to Syn-I. The RS reached the maximum iteration steps of "632 (=642 candidates-10 randomly chosen initial seeds)," while the other methods stopped within the maximum limit with

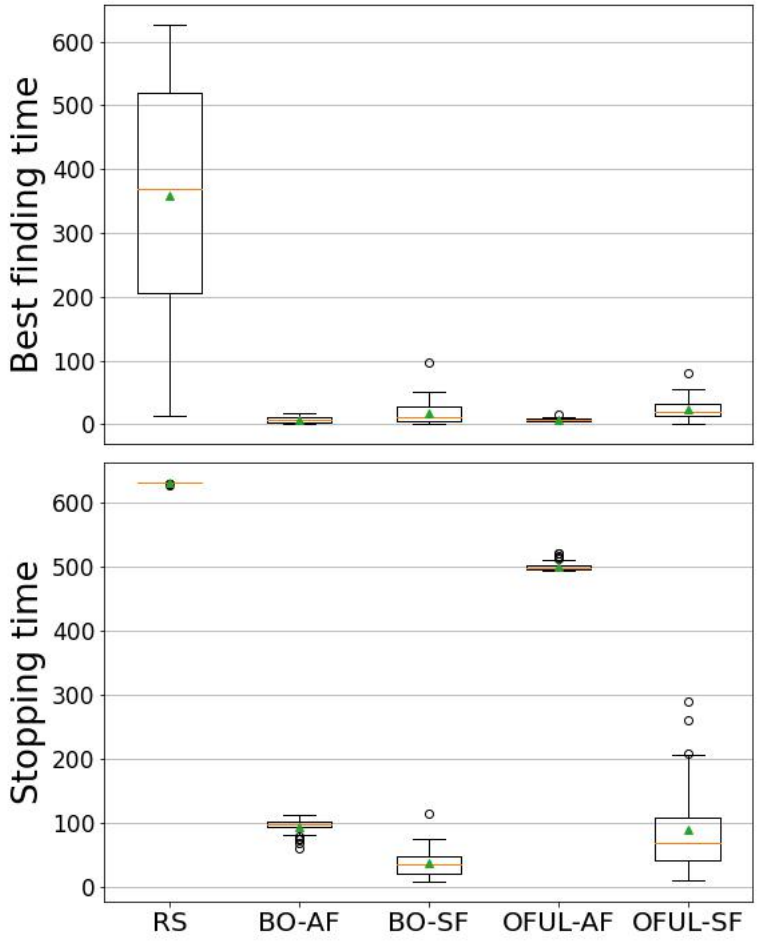

Fig. 4.2: Boxplots of best finding time (upper) and stopping time (lower) for FreeSolv dataset. The total number of molecules in FreeSolv data set is 642, and the experiments are initiated with 10 randomly chosen seeds.

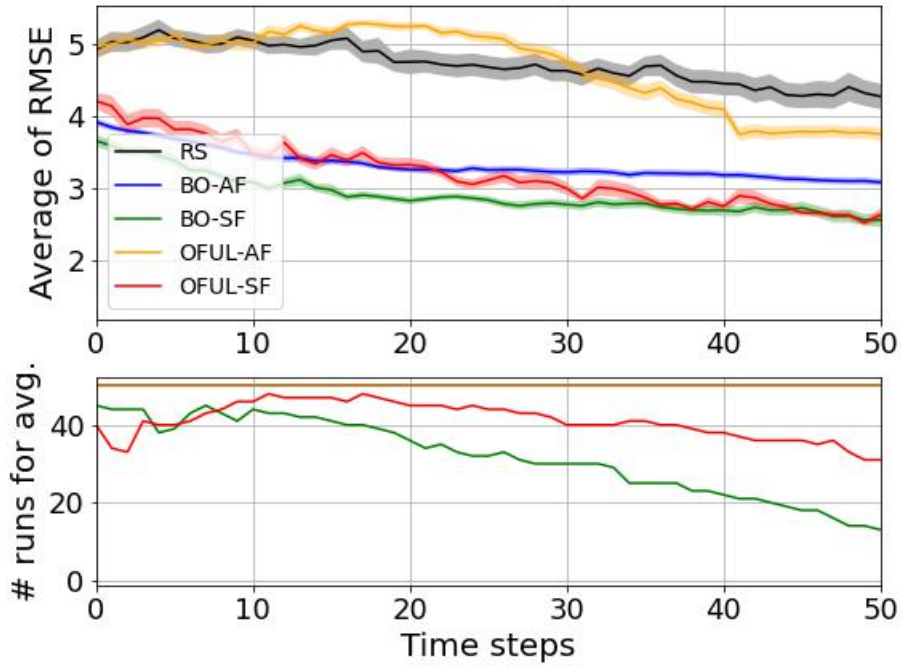

Fig. 4.3: RMSE for unobserved candidates at each time step for FreeSolv dataset. The number of runs used for statistics is also shown in the lower part

identifying the best candidate with high success rate; the success rates of BO-AF, OFUL-AF, BO-SF, and OFUL-SF were 100%, 100%, 90%, and 94%, respectively. These outcomes underscore that success rates can be diminished by feature selection dependent on the initial seeds, although feature selection accelerates the stopping process with a contraction of confidence intervals. This is interpreted that, if a region of the feature space spanned by initial seeds and those of consecutively chosen candidates is significantly different and isolated from that by the best candidate, the chosen feature variables do not necessarily predict the expected reward of the best candidate. However, note that the 90% or 94% success rate does not necessarily mean "significantly low performance" because the allowance error rate $\delta = 0.05$ allows 5% failure prediction at most within the framework of Gaussian process (BO) and linear regression model (OFUL).

The corresponding RMSE for the FreeSolv dataset was analyzed and is presented in Fig.4.3. In this datasets, the RMSEs of BO-AF, BO-SF, and OFUL-SF are significantly smaller than those of RS and OFUL-AF. The challenge encountered in regression with OFUL-AF is likely attributed to the non-linearity present in real data, which seems to be mitigated by the dynamic feature selection in OFUL-SF. Here, note that the BO-AF and BO-SF utilize a non-linear kernel, achieving better performance in RMSE. The accuracy expected from RMSE aligns well with the performances in stopping times in the FreeSolv case, resembling the Syn-I case, but not in Syn-II. In the enantioselectivity dataset, the stopping times of BO-AF and OFUL-AF are notably large due to the substantial variances in predictions and the high dimensionality in the feature space, resulting in large confidence intervals or uncertainty. These findings again underscore the correspondence of the FreeSolv and enantioselectivity datasets to Syn-I and Syn-II, respectively. Similar results were observed for Photoswitch data[48] (Explain in the **Chapter 6**).

4.4 Selected features and their influence on results

Figure 4.4 presents statistical measure of the feature selection profile of 50 runs. Here, X axis is the feature index and Y axis is the average contribution of the features. Blue and green line presents the feature contribution for success runs and unsuccess runs. And red dots and green dots are the relevant feature for ground truth catalyst. Here ground truth catalysts means catalysts having lowest hydration free energy. It is clearly seen that relevant features of the ground truth molecules has larger contribution in success runs than the unsuccess runs. However, some selected feature are not included in the ground truth.

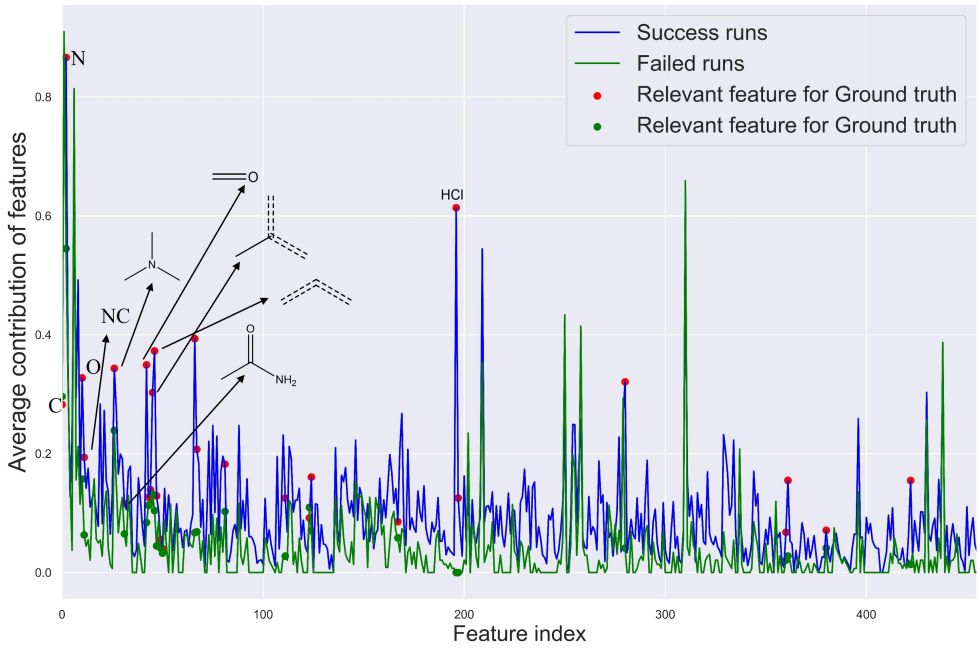

Fig. 4.4: Statistical measure of the feature selection profile of 50 runs. X axis is the feature index and Y axis is the average contribution of the features.

Average feature contribution is calculated by the following Eq. 4.1 and Eq. 4.2. The derivation are explained bellow. Feature contribution for the k^{th} run of feature j,

$$W_{k,j} = \frac{1}{N_{\text{exp}}^{(k)}} \sum_{i=1}^{N_{\text{exp}}^{(k)}} x_{i,j}^{(k)} \quad \text{where} \quad x_{i,j}^{(k)} = \begin{cases} 1, & \text{If the feature } j \text{ selected} \\ 0, & \text{Otherwise.} \end{cases} \tag{4.1}$$

Their averages over success runs $N^{(\text{suc})}$

$$W_{k,j} = \frac{1}{N^{(\text{suc})}} \sum_{k \in \text{success}} W_{k,j} \tag{4.2}$$

Likewise, feature contribution for the failure runs are also calculated.

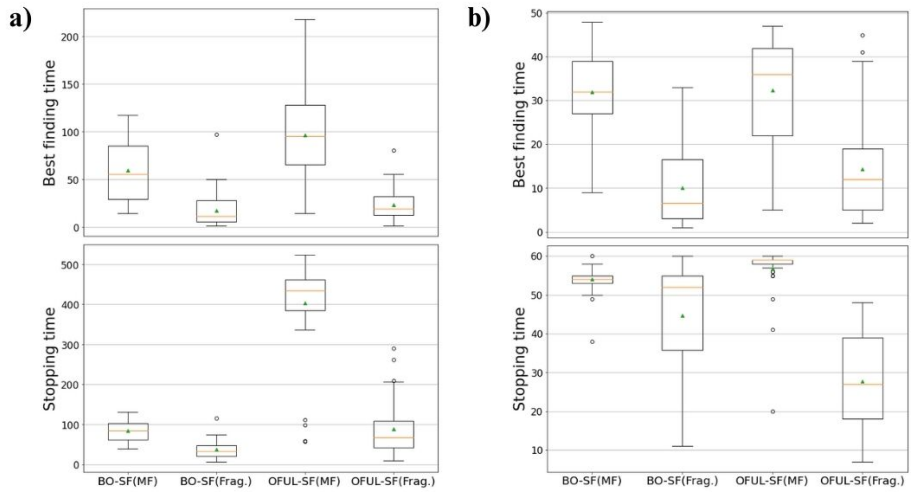

Fig. 4.5: The boxplots for best finding time and stopping time for (a) FreeSolv data, and (b) Enantioselectivity data using Morgan fingerprints feature (MF) and fragment descriptor (frag.) with BO-SF and OFUL-SF schemes. The meaning of of each symbols is the same as in Fig.2. For example, for FreeSolv data, the median values of best finding times are 55.0, 95.0, 11.0, and 19.0 for BO-SF(MF), OFUL-SF(MF), BO-SF (frag.), and OFUL-SF (frag.), respectively.

4.5 Analysis with Morgan fingerprint descriptor

The results were evaluated using the Morgan fingerprint (MF) descriptor[62,63] and compared with those obtained using the Fragment (frag.) descriptor[56,58] for FreeSolv[46] and Enantioselectivity[47] data. A detailed analysis of the Enantioselectivity data will be explained in **Chapter 5**. Result of Enantioselectivity is included here solely for comparison purpose. Readers are encouraged to return to this section after reading **Chapter 5**. Here the conditions except the descriptor was set to be same.

Fig. 4.5 (a) and (b) present the boxplots of the best finding time statistics and the stopping time statistics for FreeSolv data and Enantioselectivity data, respectively. As the overall trend, one can see Morgan fingerprints feature require significantly longer time in both times than the fragment descriptor. The success rates, i.e.,—how many percentage the underlying target molecule (the

best molecule having either lowest hydration free energy or highest enantioselectivity) are detected—, for FreeSolv data are 42.0% and 94% for the BO-SF(MF) and OFUL-SF(MF), while those are 90% and 94% for BO-SF(frag.) and OFUL-SF(frag.) respectively. Likewise, the success rates for Enantioselectivity data are 100% and 74% for the BO-SF(MF) and OFUL-SF(MF), while those are 100% and 98% for BO-SF(frag.) and OFUL-SF(frag.), respectively. The difference between Morgan fingerprints and ISIDA (In Silico Design and Data Analysis) fragment descriptor are as follows: the former takes into account solely whether the substructure present or absent within a defined radius (binary vector),[57] whereas the latter does how many substructures present within a define radius (it counts the number of fragments).[59,47] There are also some distinct advantages of fragment descriptor features such as the ability to uniquely represent any molecular graph invariant, express any symmetric similarity measure, and facilitate the representation of regression or classification models.[59] These could be the reason why the difference exists between the two feature variables. However, it may depend on the system to be studied and further investigation is required for concluding which descriptor is better to use in general.

4.6 Kernel comparison for Bayesian optimization

In this research RBF kernel was employed for Bayesian optimization. The results of the Matérn 5/2 kernel are presented here for comparison with those of the RBF kernel, using the fragment descriptor. The other conditions are same to those in Fig. 4.2. Fig. 4.6(a) and (b) shows the results for FreeSolv, and Enantioselectivity data, respectively, using BO-AF and BO-SF with two different kernels, Matérn 5/2 and RBF. The success rates of BO-SF (Matérn 5/2) and

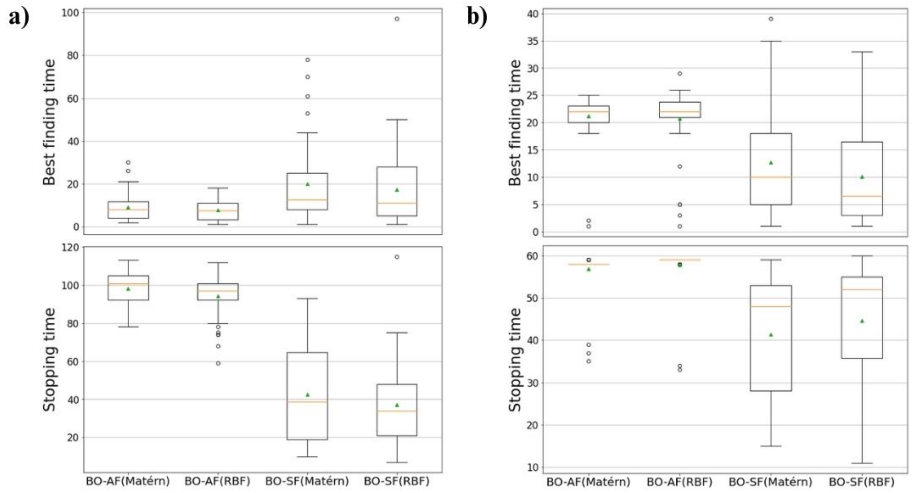

Fig. 4.6: The boxplot of best finding time and stopping time evaluated by Bayesian optimization with Matérn 5/2 kernel and RBF kernel for a) FreeSolv data and b) Enantioselectivity data. The meaning of each symbols is the same as in Fig. 2.

BO-SF (RBF) is 88.0% and 90.0%, respectively, and those of both BO-AF are 100% for FreeSolv data. For enantioselectivity data, those of BO-AF (Matérn 5/2) is 98.0% whereas those of all the others are 100%. This shows that the performances of Matérn 5/2 kernel and RBF kernel are almost the same at least for the current systems. (Details analysis of the Enantioselectivity data will be explained in the **Chapter 5**. Result of Enantioselectivity is included here solely for comparison purpose. Readers are encouraged to return to this section after reading **Chapter 5**)

4.7 Application of Tanimoto kernel for Bayesian optimization

The Bayesian Optimization (BO) was recomputed using the Tanimoto kernel for Morgan fingerprint descriptors. The current results are presented in Fig. 4.7, based on 50 runs. Median best (molecule) finding times are 26.0, 49.0, and

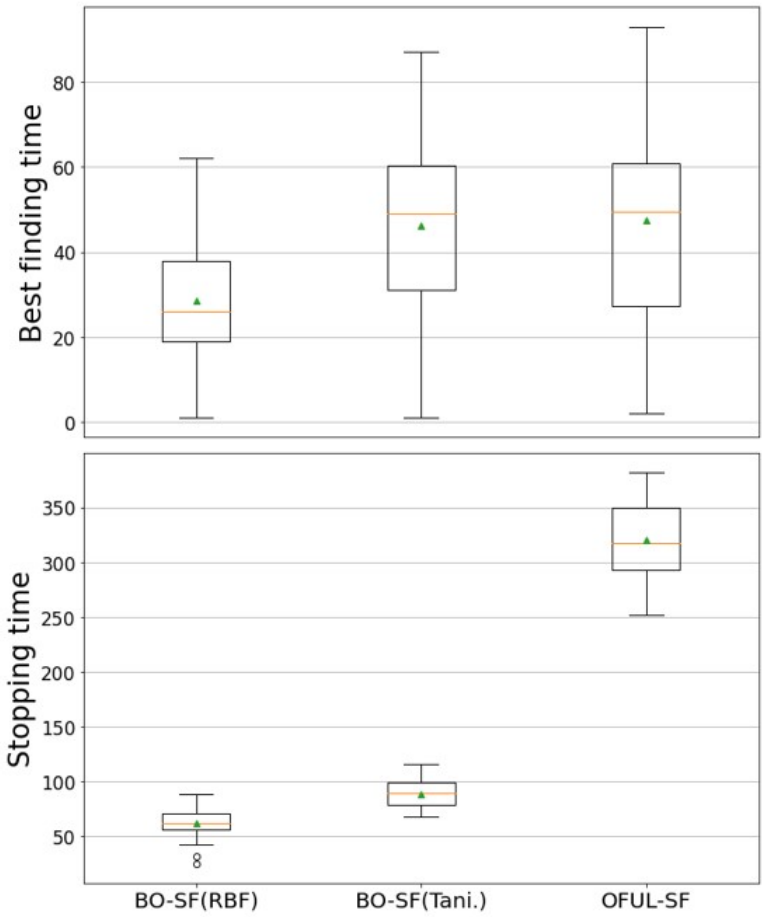

Fig. 4.7: Comparison of best finding time and stopping along BO-SF(RBF), BO-SF(Tani.) and OFUL-SF with Morgan fingerprint features. (Note, Tani.: Tanimoto kernel; RBF: Radial basis function.)

49.5, and median stopping times are 62.0, 89.0, and 317.5 for BO-SF(RBF), BO-SF(Tani.) and OFUL-SF. It was confirmed that the Tanimoto kernel can also be used for Morgan fingerprint descriptors. Since RBF kernel has been demonstrated to work better than the Tanimoto kernel and is easier to apply, the RBF kernel is used for all the data. (Note: RBF refers to the radial basis function kernel, and Tani. refers to the Tanimoto kernel.)

4.8 Choice of allowable noise δ

The allowed error rate (δ) for the model which determine the maximum probability of the $\hat{\theta}_t$ not being inside of the confidence ellipsoid presented in equation (2.6) can be selected by striking a balance between failure rate and stopping time. The definitions of failure rate and stopping time are explained in the *Brief Discussion of Methods* section in **Chapter 3** as well as in the *Stopping Condition* section in **Chapter 2**. The model's error rate is inversely related to stopping time and directly related to the failure rate. The following Fig. 4.8 establishes

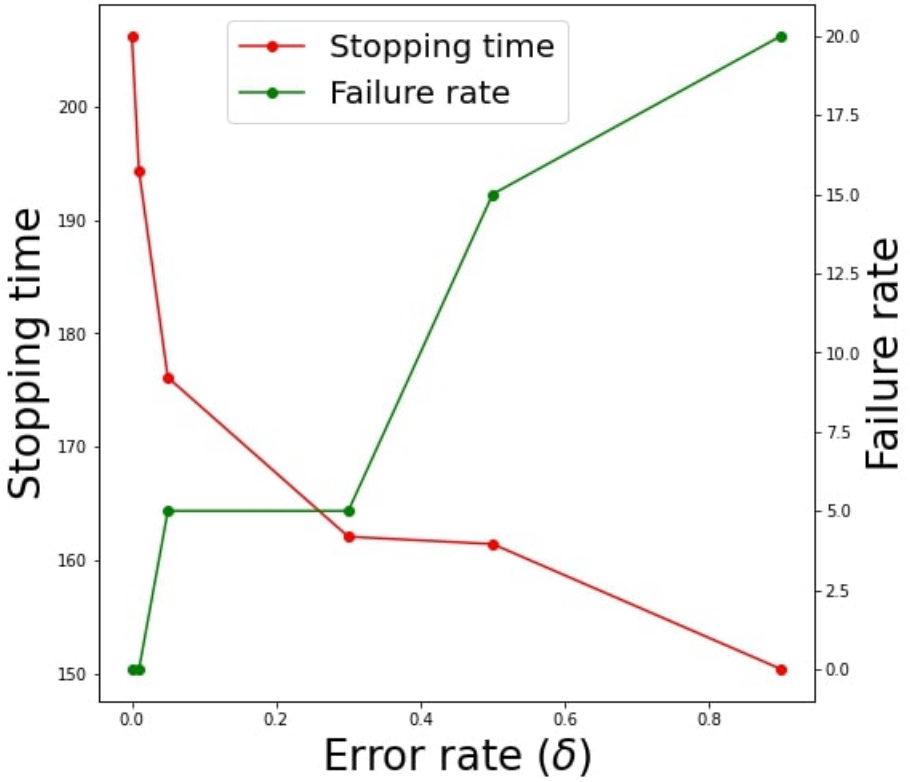

Fig. 4.8: Tradeoff between stopping time and failure rate through the change of error rate δ.

the relationship among the quantities. The researcher has the flexibility to assess the error rate by considering either or both the stopping time and the failure rate, depending on their specific focus or preference. The above result is constructed

from the Syn-II data with 20 independent runs.

4.9 Parameter optimization for RBF kernel

The Bayesian Optimization (BO) method was used for comparison with the linear bandit algorithm. Gaussian process regression with radial basis function (RBF) kernel is used as surrogate function and the Eq. 2.27 in the main text is considered as acquisition function. Note that, **surrogate function** is a simplified model used to approximate and optimize a complex and expensive objective function efficiently and **acquisition function** is a strategy used in optimization to balance exploration and exploitation by guiding the selection of the next points to evaluate in the search space.

One important parameter for Gaussian process regression is length scale parameter which determines the smoothness of the model. This parameter should be optimized in the model fitting process. Widely used python package, Scikit-learn is utilized to fit the BO model where default length scale parameter range $\left(10^{-5}, 10^{5}\right)$ is suggested for RBF kernel. But this large range shows some prior like prediction. Here prior like prediction means, prediction of unobserved data is zero values. Figure 4.9 presents some example of the prior like prediction though the prediction of observed data is satisfactory. It is observed that this problem occurs when the length scale parameter is set to very large or very small values. These extreme values make it difficult to understand the data pattern, necessitating a data-driven range to address the issue. Therefore, it is suggested that a comparatively small range be used to optimize the length scale parameter instead of the default range $\left(10^{-5}, 10^{5}\right)$ which will be derived from data. The new range can be constructed using the following steps:

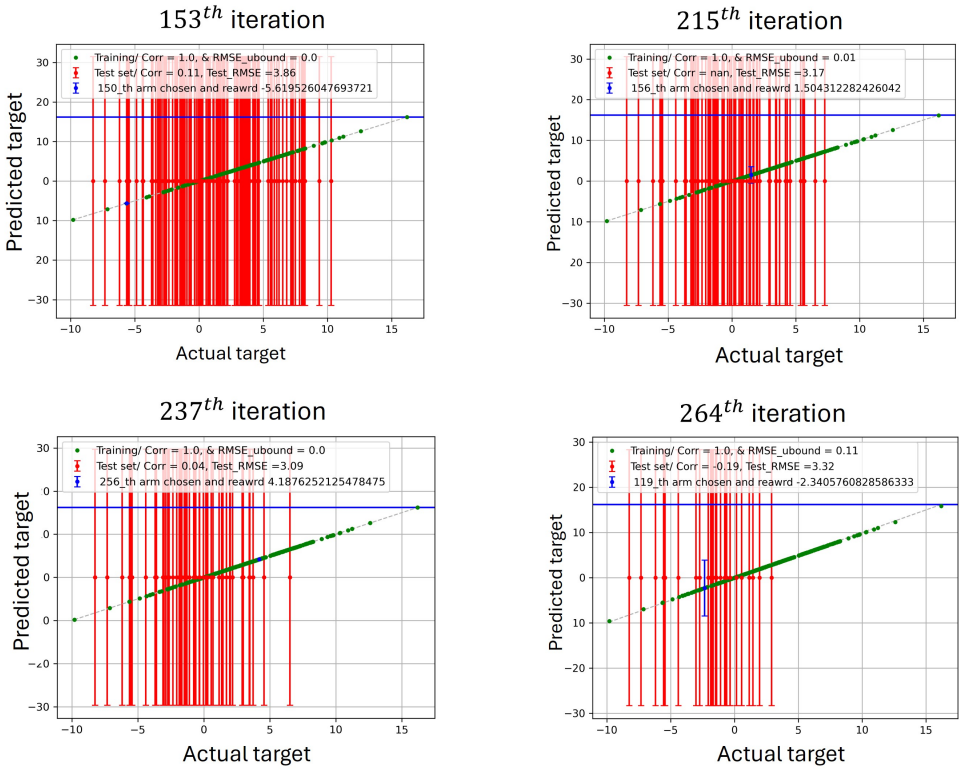

Fig. 4.9: Example of prior like prediction for different iteration steps. This figures are constructed using Syn-II data. In the Syn-II data this types of predictions are observed many times. It is clearly visible that the prediction for training data is very good but for test data prediction is close to zero. This poor prediction also shown by RMSE and Correlation value in the legend.

1. Calculate the pairwise Euclidean distance $D = \{d_{x_i,x_j} \mid i \neq j\}$ where x_i and x_j are i^{th} and j^{th} observed samples/ molecules in this research .

2. Calculate the minimum and maximum of the distance values as $m = \min(D)$ and $M = \max(D)$ respectively.

3. Construct length scale range as (m, M) and optimize the length scale parameter within this range.

4.10 Conclusion

In this chapter, the FreeSolv data and the application of the newly developed algorithm are explained. It is observed that the OFUL bandit algorithm is comparable to the traditionally used Bayesian optimization method. Additionally, comparisons of kernels, kernel parameter optimization, and a discussion on the Tanimoto kernel are provided.

Chapter 5

Sequential Optimization of Experimental Data of Enantioselectivity

5.1 Brief discussion of methods

The linear bandit algorithm (OFUL bandit algorithm) and Bayesian optimization (BO) along with their stopping condition have been discussed in the **Chapter 2**. A brief discussion of this topic was provided in **Chapter 3**. To save time and space, the same discussion is excluded from this chapter.

5.2 Enantioselectivity data

Enantiomer refers the chiral molecules those are the mirror image of each other and are not superimposed. Lets see the definition of chiral molecules, it is defined as the molecule that have nonsuperimposable property on its mirror image. Some molecules are not mirror images of one another and are not super-

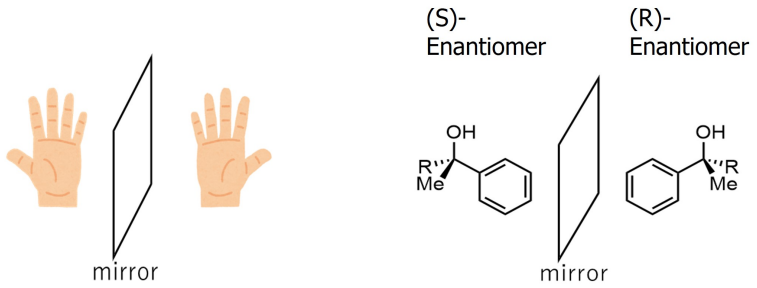

Fig. 5.1: Example of enantiomers. How the mirror image used to express the enantiomers presented in right figure.

imposable those are known as diastereomers. Enantiomers are identical chemical symbol but express different chemical structure also often shows chemical properties. Figure 5.1 shows the mirror image of hand for reference to understand the enantiomers (left figure), and corresponds to the (S)-enantiomer and (R)-enantiomer of —-chemical name—(right figure). The enantioselectivity is measured by the difference between the free energy barriers of the enantiomers $\Delta\Delta G$. The difference between the free energy barriers of the enantiomers is defined as follows $\Delta\Delta G$. Figure 5.2 shows that reactant (in the middle of the figure) can produce two product (S)-enantiomer (left of the figure) and (R)-enantiomer (right of the figure), if the reactant can surpass $\Delta G^{\ddagger}(S)$ energy barrier then it produce (S)-enantiomer and if surpass $\Delta G^{\ddagger}(R)$ energy barrier then it produce (R)-enantiomer. Now the enantiomer selection measured by the expression $\Delta\Delta G = \Delta G^{\ddagger}(S) - \Delta G^{\ddagger}(R)$. Larger the difference exhibits an unequal production ratio. The goal is to identify the larger absolute values of $\Delta\Delta G$ i.e. $|\Delta\Delta G|$. In the figure production ratio of (R)-enantiomer is larger than (S)-enantiomer because its energy barrier is smaller. The enantioselectivity data is extracted from the work conducted by Tsuji et. al.[47] This data is consisted of 70 reaction systems.

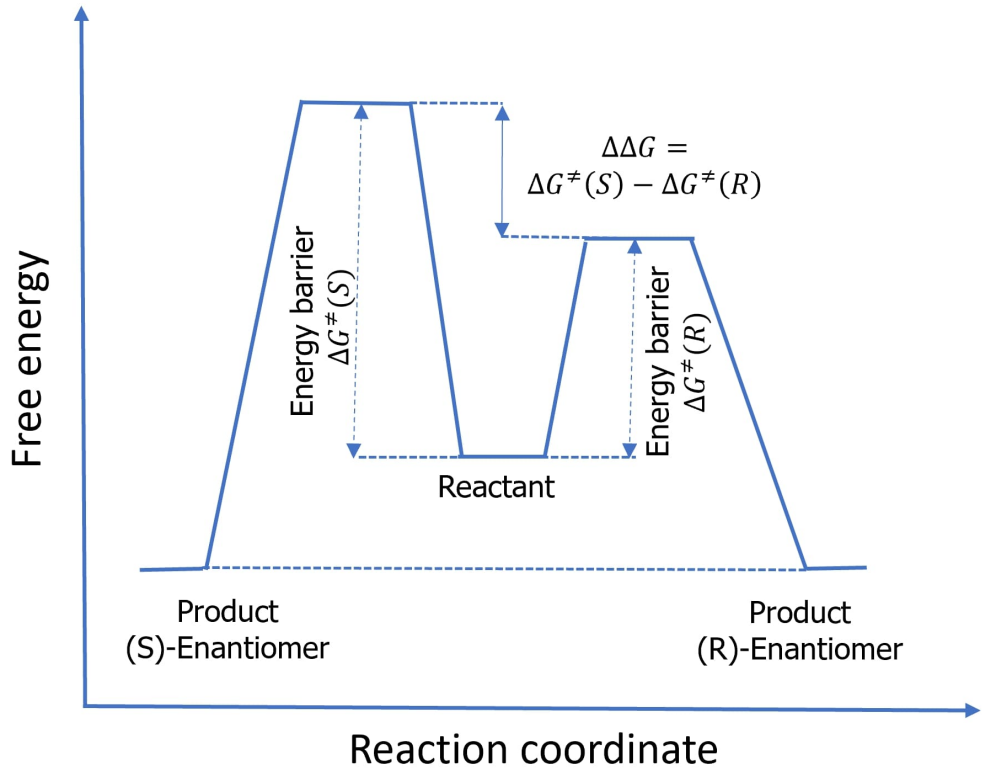

Fig. 5.2: Definition of enantioselectivity.

5.3 Enantioselectivity data analysis

Figure 5.3 presents boxplots of the best finding time and stopping time for enantioselectivity data. The median values for the best finding times are 33.0, 22.0, 6.5, 27.0, and 12.0 for RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, respectively. As for stopping times, the median values are 60.0, 59.0, 52.0, 60.0, and 27.0 for RS, BO-AF, BO-SF, OFUL-AF, and OFUL-SF, respectively. The trend in best finding times and stopping times aligns more closely with Syn-II (presented in **Chapter 3**), as opposed to the FreeSolv case (presented in **Chapter 4**). Feature selection analysis using the full candidate data was conducted for both FreeSolv and enantioselectivity datasets, revealing that FreeSolv has 95 relevant feature variables, constituting 21% of the total, while the enantioselec-

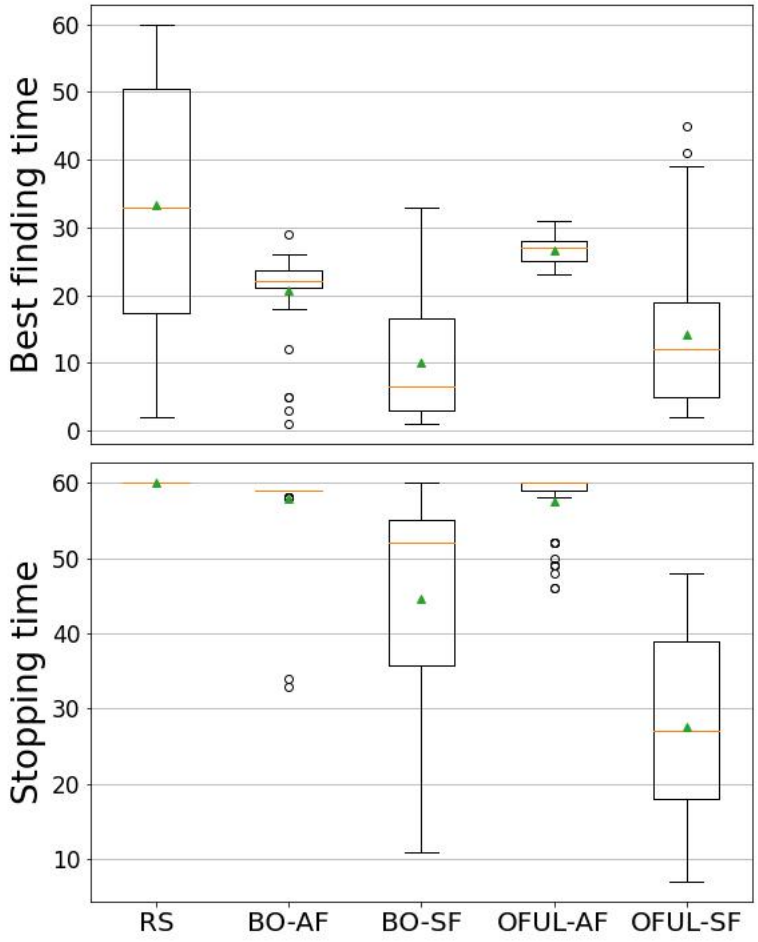

Fig. 5.3: Boxplots of best finding time (upper) and stopping time (lower) for Enantioselectivity dataset. The total number of molecules in enantioselectivity dataset is 70, and the experiments are initiated with 10 randomly chosen seeds.

tivity dataset has 15 relevant feature variables, representing 10%. Each value of percentage reasonably corresponds to 33% of Syn-I and 6% of Syn-II cases, respectively, which could be one factor in understanding the characteristics of the results because the ratio of relevant features is related to both the regression model and the construction of the confidence interval. Only OFUL-SF achieved successful stopping with a 98% success rate, while all other methods reached the maximum iteration limit of "60 (=70 candidates - 10 randomly chosen initial seeds)". The OFUL-SF demonstrated an average reduction of experimental cost by two-thirds, significantly contributing to the exploration of new catalysis for desired enantioselective reactions, given the substantial human cost associated with catalyst preparation. In contrast to the Syn-II case(presented in **Chapter 3**), where BO-SF tends to stop faster than OFUL-SF, in the present case the OFUL-SF can stop significantly faster than BO-SF, suggesting a complementary property between BO and OFUL depending on the target system.

The RMSE for each of the models for the enantioselectivity datasets was analyzed, as presented in Figure 5.4. The RMSEs of BO-AF, BO-SF, and OFUL-SF are significantly smaller than those of RS and OFUL-AF. The challenge encountered in regression with OFUL-AF is likely attributed to the non-linearity present in real data, which seems to be mitigated by the dynamic feature selection in OFUL-SF. Here, note that the BO-AF and BO-SF utilize a non-linear kernel, achieving better performance in RMSE. The accuracy expected from RMSE aligns well with the performances in stopping times in the FreeSolv case(**Chapter 4**), resembling the Syn-I case, but not in Syn-II. In the enantioselectivity dataset, the stopping times of BO-AF and OFUL-AF are notably large due to the substantial variances in predictions and the high dimensionality in the feature space, resulting in large confidence intervals or uncertainty. These find-

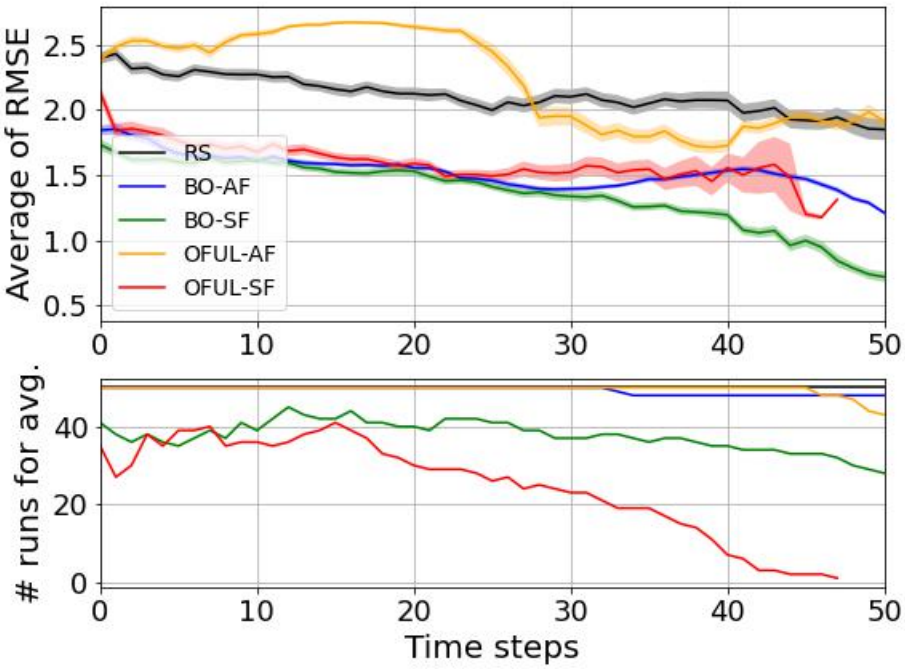

Fig. 5.4: RMSE for unobserved candidates at each time step for Enantioselectivity dataset. The number of runs used for statistics is also shown in the lower part. The lines corresponding to RS, BO-AF and OFUL-AF for FreeSolv data are overlapped due to the unstopped runs until time steps 50.

ings again underscore the correspondence of the FreeSolv and enantioselectivity datasets to Syn-I and Syn-II, respectively.

5.4 Selected features and their influence on results

Figure 5.5 presents statistical measure of the feature selection profile of 50 runs. Here, X axis is the feature index and Y axis is the average contribution of the features. Blue and green line presents the feature contribution for success run and unsuccess run. And red dots and green dots are the relevant feature for ground truth catalyst. Here ground truth catalysts means catalysts having highest enantioselectivity. It is clearly seen that relevant features of the ground truth molecules has larger contribution in success runs than the unsuccess runs. However, some selected feature are not included in the ground truth.

Feature contribution is calculated by the Eq. 4.1 and Eq. 4.2 stated in **Chapter 4**.

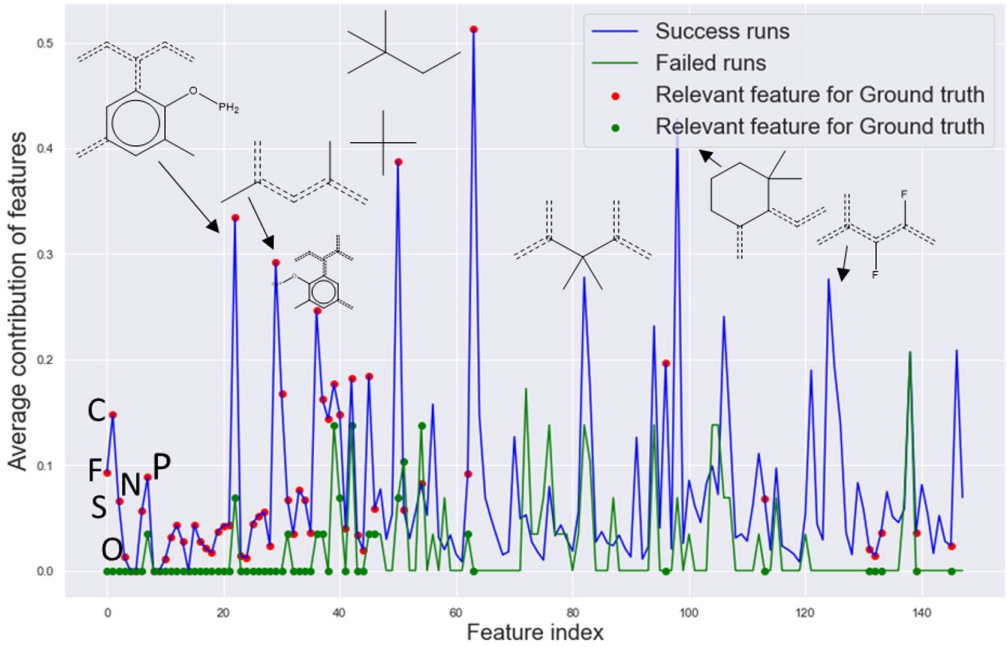

Fig. 5.5: Statistical measure of the feature selection profile of 50 runs. X axis is the feature index and Y axis is the average contribution of the features.

5.5 Conclusion

In this chapter, the focus has been on sequential optimization of enantioselectivity data, with the goal of identifying the catalyst with the highest selectivity. Selectivity is measured based on the energy barrier difference between two enantiomers denoted as $\Delta\Delta G = \Delta G^{\ddagger}(S) - \Delta G^{\ddagger}(R)$, where $\Delta G^{\ddagger}(S)$ and $\Delta G^{\ddagger}(R)$ are energy barrier to produce S enantiomer and R enantiomer, respectively. It is observed that the OFUL bandit algorithm is comparable to the traditionally used Bayesian optimization method.

Chapter 6

Sequential Optimization of Experimental Data of Photoswitch

6.1 Photoswitch data

In brief, the Photoswitch data data consists of 405 molecules and 4 target variables, which are used for multi-objective optimization.[49,48] In this research, the target variable of "E isomer $\pi - \pi^*$ Transition wavelength" among four different target variables because of its largest number of the available target values have been chosen. The data includes 392 molecules with 298 features of the fragment descriptor. The problem we set for this data is to find the largest wavelength in a few experiment as possible.

6.2 Brief discussion of methods

The linear bandit algorithm (OFUL bandit algorithm) and Bayesian optimization (BO) along with their stopping condition have been discussed in the **Chapter 2**. A brief discussion of this topic was provided in **Chapter 3**. To save time

and space, the same discussion is excluded from this chapter.

6.3 Photoswitch data analysis

To demonstrate the performance of OFUL and BO Photoswitch[48,49] dataset have been studied in addition to the synthetic data, FreeSolv, and Enantioselectivity data. The target variable "E isomer $\pi - \pi^*$ transition wavelength" was chosen among four different target variables due to its having the largest number of available target values. The data includes 392 molecules with 298 features of the fragment descriptor. The other conditions of computations are set to be the same to those in Fig. 4.2. Fig. 6.1 shows the results of BO-SF and OFUL-SF. The median values of best finding times are 26.0 and 49.5 for BO-SF and OFUL-SF. The median values of stopping times are 62.0 and 317.5 for BO-SF and OFUL-SF. The success rates are 100% for both methods. The result follows a similar pattern of FreeSolv data. Note that the current version of OFUL is based on the simplest, linear regression while BO adopts nonlinear dependence between the target variable and the feature variables. In the formula of linear regression (Eq. (1)) non-linearity can be incorporated with replacing x_i to $f_i(\{x_i\})$. A systematic scrutiny to incorporate non-linearity is one of the forthcoming subjects to be addressed along OFUL-SF.

The RMSE for each of the models for the photoswitch datasets was analyzed, as presented in Figure 6.2. The RMSEs of BO-AF, BO-SF, and OFUL-SF are significantly smaller than those of RS and OFUL-AF, i.e, the average RMSE and Number of unstopped runs for BO-AF, BO-SF and OFUL-SF decrease faster than others. The challenge encountered in regression with OFUL-AF is likely attributed to the non-linearity present in real data, which seems to

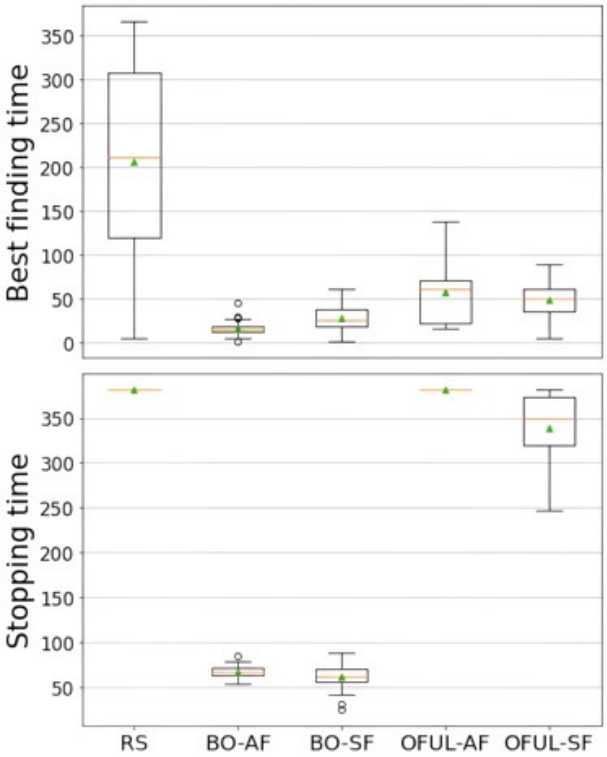

Fig. 6.1: The boxplot of best finding time and stopping time evaluated by BO-SF and OFUL-SF for Photoswitch data. The meaning of each symbols is the same as in Fig.4.2.

be mitigated by the dynamic feature selection in OFUL-SF. Here, note that the BO-AF and BO-SF utilize a non-linear kernel, achieving better performance in RMSE.

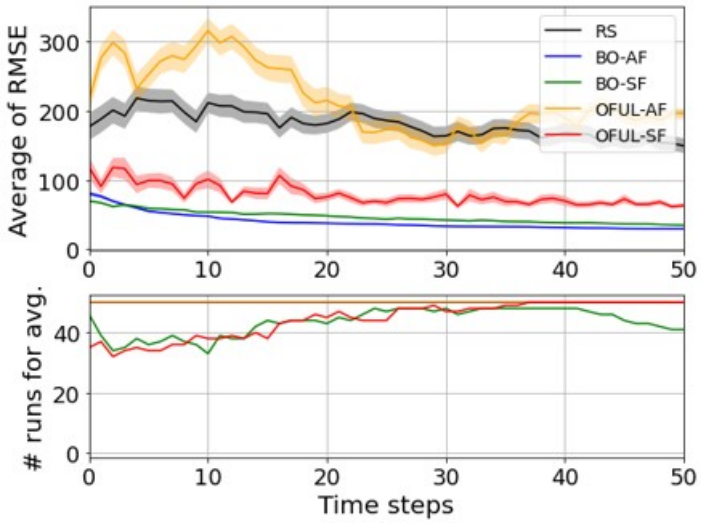

Fig. 6.2: RMSE for unobserved candidates at each time step for Photoswitch dataset. The number of runs used for statistics is also shown in the lower part.

6.4 Selected features and their influence on results

Figure 6.3 presents statistical measure of the feature selection profile of 50 runs. Here, X axis is the feature index and Y axis is the average contribution of the features. Blue line presents the feature contribution for success run. And red dots are the relevant feature for ground truth catalyst. Here ground truth catalysts means catalysts having highest wavelength. It is clearly seen that relevant features of the ground truth molecules has larger contribution in success runs. In contrast to FreeSolv data or Enantioselectivity data unsuccess case is not shown because success rate of Photoswitch data is 100%. However, some selected feature are not included in the ground truth.

Feature contribution is calculated by the Eq. 4.1 and Eq. 4.2 stated in **Chapter 4**.

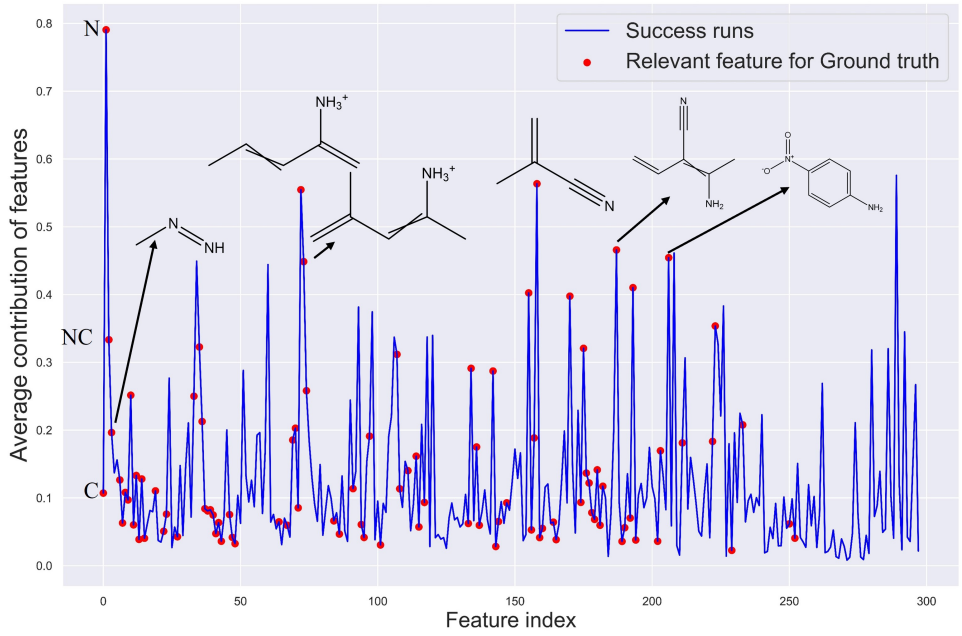

Fig. 6.3: Statistical measure of the feature selection profile of 50 runs. X axis is the feature index and Y axis is the average contribution of the features.

6.5 Conclusion

This chapter discusses on sequential optimization of the photoswitch data, with the goal of identifying the largest wavelength molecule within fewer experiment as possible. It is observed that the OFUL bandit algorithm is comparable to the traditionally used Bayesian optimization method.

Chapter 7

General Conclusion

7.1 Conclusion

A sequential optimization is one of the promising approaches in identifying the optimal candidate(s) (molecules, reactants, drugs etc.) with desired properties (such as drug efficacy, reaction yield, enantioselectivity, wavelength and solvation affinity etc.) from a large set of potential candidates, while minimizing the number of experiments is required. In this research, a new sequential optimization algorithm for molecular problems have been developed based on a reinforcement learning framework, specifically the linear multi-armed bandit approach (Optimism in the Face of Uncertainty Linear-OFUL bandit). This algorithm incorporates online, dynamic feature selection, where relevant molecular descriptors are updated throughout the experiments. In this algorithm ISIDA ("In Silico Design and data Analysis) fragment descriptors x are calculated from the structure of the molecule are used as the feature for the prediction of the molecular property y termed as target through a linear regression. The molecular structure is constructed so called Simplified Molecular-Input Line-Entry System

(SMILES) codes. A distinctive feature of our algorithm is its ability not only to identify the optimal candidate molecule but also to automatically determine when to stop the search based on the predictions for candidate molecules and their confidence intervals evaluated by the regression models. This algorithm design in a way where we sequentially chose one molecule by considering the upper confidence bound of the prediction and observe the reward (molecular property). The stopping condition is then checked to determine when the algorithm should be stopped. This condition is designed to ensure the reliability of the chosen candidate from the data set pool. The stopping condition is defined using the maximum of the upper confidence bound and the maximum of the observed reward. If the former quantity is larger than or equal to the latter, the algorithm continues; if it is smaller, the algorithm stops.

However, the high dimensionality of the feature space (e.g., molecular fragment descriptors, Morgan fingerprint descriptors) makes it often difficult to utilize the relevant features during the process of updating the set of candidates to be examined so that we incorporate feature selection method, LASSO (Least Absolute Shrinkage and Selection Operator) for each of the sequential iteration consequently features space dynamically changed and reduced. It was found that dynamic feature selection in representing the desired properties throughout the experiments provides better performance, such as reducing the time required to find the best candidate and to stop the experiment. Overall, the multi-armed linear bandit approach with a dynamic feature selection scheme (abbreviated as OFUL-SF, also OFUL-AF exists where all feature used) outperformed the standard Bayesian Optimization (BO) with fixed feature variables.

The method underwent validation, starting with synthetic datasets based on linear models and extending to real datasets, including benchmark hydration

free energy dataset, experimental data of enantioselective reactions and Photoswitch dataset. Comparative analyses against the linear bandit algorithm without feature selection, Bayesian optimization (BO) with and without feature selection, and random search affirmed the essential role of dynamic feature selection, particularly in datasets with numerous irrelevant feature variables included in the total feature variables. It is found the critical role of feature selection in enhancing predictive performance and reducing confidence intervals during sequential optimization, thereby accelerating the automatic judgment capabilities of the algorithm in stopping the search. While the performance of our algorithm and Bayesian optimization with feature selection proved comparable, our algorithm demonstrated superior performance in challenging real-world applications, particularly in the case of enantioselective reactions.

7.2 Limitation

This research doesn't include the chemical interpretation of the features. This interpretation is extensively important to make sense how the features are responsible to reflect the molecular properties. This is very tough job indeed.

7.3 Outlook

In a linear bandit setting, the expected reward is modeled as a linear combination of features associated with each action. Extending it to a non-linear bandit allows for more complex relationships between actions and rewards, capturing non-linear patterns in the data. By extending linear bandit to non-linear bandit some benefits may achieved such as improved modeling, increased accuracy, better decision making and enhanced exploration-exploitation trade-off etc. It's

important to note that the benefits may vary depending on the specific problem and dataset. While non-linear bandits offer advantages in capturing more complex relationships, they also introduce challenges in terms of model complexity and interpretability. The choice of the non-linear model should be guided by the characteristics of the problem at hand and the available data.

Chemical interpretation of the selected feature can be studied to understand how the molecular properties are reflected by molecular features.

References

[1] Stephen Gow, Mahesan Niranjan, Samantha Kanza, and Jeremy G Frey. A review of reinforcement learning in chemistry. *Digital Discovery*, 1:551–567, 2022.

[2] Tomohiro Nakamura, Shinsaku Sakaue, Kaito Fujii, Yu Harabuchi, Satoshi Maeda, and Satoru Iwata. Selecting molecules with diverse structures and properties by maximizing submodular functions of descriptors learned with graph neural networks. *Sci. Rep.*, 12(1):1124, 2022.

[3] Peter Kirkpatrick and Clare Ellis. Chemical space. *Nature*, 432(7019):823–824, 2004.

[4] Meena K. Sakharkar, Karthic Rajamanickam, Chidambaram S. Babu, Jitender Madan, Ramesh Chandra, and Jian Yang. Preclinical: Drug target identification and validation in human. In Shoba Ranganathan, Michael Gribskov, Kenta Nakai, and Christian Schönbach, editors, *Encyclopedia of Bioinformatics and Computational Biology*, pages 1093–1098. Academic Press, Oxford, 2019.

[5] A Patrícia Bento, Anne Hersey, Eloy Félix, Greg Landrum, Anna Gaulton, Francis Atkinson, Louisa J Bellis, Marleen De Veij, and Andrew R Leach.

An open source chemical structure curation pipeline using rdkit. *J Cheminf*, 12:1–16, 2020.

[6] Nobuaki Kikkawa and Hiroshi Ohno. Materials discovery using max k-armed bandit. *arXiv preprint arXiv:2212.08225*, 2022.

[7] Adrià Pérez, Pablo Herrera-Nieto, Stefan Doerr, and Gianni De Fabritiis. Adaptivebandit: a multi-armed bandit framework for adaptive sampling in molecular simulations. *J. Chem. Theory Comput.*, 16(7):4685–4693, 2020.

[8] Dario Caramelli, Jarosław M Granda, S Hessam M Mehr, Dario Cambié, Alon B Henson, and Leroy Cronin. Discovering new chemistry with an autonomous robotic platform driven by a reactivity-seeking neural network. *ACS Cent. Sci.*, 7(11):1821–1830, 2021.

[9] Gheorghe Lisca, Daniel Nyga, Ferenc Bálint-Benczédi, Hagen Langer, and Michael Beetz. Towards robots conducting chemical experiments. In *2015 IEEE/RSJ Int. Conf. on IROS*, pages 5202–5208. IEEE, 2015.

[10] Connor W Coley, Dale A Thomas III, Justin AM Lummiss, Jonathan N Jaworski, Christopher P Breen, Victor Schultz, Travis Hart, Joshua S Fishman, Luke Rogers, Hanyu Gao, et al. A robotic platform for flow synthesis of organic compounds informed by ai planning. *Science*, 365(6453):eaax1566, 2019.

[11] Ke Wang and Alexander W Dowling. Bayesian optimization for chemical products and functional materials. *Curr. Opin. Chem. Eng.*, 36:100728, 2022.

[12] Edward O Pyzer-Knapp. Bayesian optimization for accelerated drug discovery. *IBM J. Res. Dev.*, 62(6):2:1–2:7, 2018.

[13] Tsuyoshi Ueno, Trevor David Rhone, Zhufeng Hou, Teruyasu Mizoguchi, and Koji Tsuda. Combo: An efficient Bayesian optimization library for materials science. *Mater. Discov.*, 4:18–21, 2016.

[14] Jiahao Yan, Han Wei, Han Xie, Xiaokun Gu, and Hua Bao. Seeking for low thermal conductivity atomic configurations in sige alloys with bayesian optimization. *ESEE*, 8(4):56–64, 2020.

[15] Mahdi Imani and Seyede Fatemeh Ghoreishi. Bayesian optimization objective-based experimental design. In *ACC*, pages 3405–3411. IEEE, 2020.

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

[17] Jose Antonio Garrido Torres, Sii Hong Lau, Pranay Anchuri, Jason M Stevens, Jose E Tabora, Jun Li, Alina Borovika, Ryan P Adams, and Abigail G Doyle. A multi-objective active learning platform and web app for reaction optimization. *J. Am. Chem. Soc.*, 144(43):19999–20007, 2022.

[18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[19] Piotr S Gromski, Alon B Henson, Jarosław M Granda, and Leroy Cronin. How to explore chemical space using algorithms and automation. *Nat. Rev. Chem.*, 3(2):119–128, 2019.

[20] D Balamurugan, Weitao Yang, and David N Beratan. Exploring chemical space with discrete, gradient, and hybrid optimization methods. *J. Chem. Phys.*, 129(17), 2008.

[21] Jean-Louis Reymond and Mahendra Awale. Exploring chemical space for drug discovery using the chemical universe database. *ACS Chem. Neurosci.*, 3(9):649–657, 2012.

[22] David Eriksson and Martin Jankowiak. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

[23] Kristian Kersting, Christian Plagemann, Patrick Pfaff, and Wolfram Burgard. Most likely heteroscedastic gaussian process regression. In *ICML*, pages 393–400, 2007.

[24] Ryan-Rhys Griffiths, Alexander A Aldrick, Miguel Garcia-Ortegon, Vidhi Lalchand, et al. Achieving robustness to aleatoric uncertainty with heteroscedastic bayesian optimisation. *MLST*, 3(1):015004, 2021.

[25] Anastasia Makarova, Ilnura Usmanova, Ilija Bogunovic, and Andreas Krause. Risk-averse heteroscedastic bayesian optimization. *NIPS*, 34:17235–17245, 2021.

[26] Alexander I Cowen-Rivers, Wenlong Lyu, Rasul Tutunov, Zhi Wang, Antoine Grosnit, Ryan Rhys Griffiths, Alexandre Max Maraval, Hao Jianye, Jun Wang, Jan Peters, et al. HEBO: Pushing the limits of sample-efficient hyperparameter optimisation. *JAIR*, 74:1269–1349, 2022.

[27] Juan Maroñas, Oliver Hamelijnck, Jeremias Knoblauch, and Theodoros Damoulas. Transforming gaussian processes with normalizing flows. In *International Conference on Artificial Intelligence and Statistics*, pages 1081–1089. PMLR, 2021.

[28] Eyal Even-Dar, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *JMLR*, 7(6), 2006.

[29] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In 23^{rd} *COLT*, pages 41–53, 2010.

[30] Koji Tabata, Atsuyoshi Nakamura, Junya Honda, and Tamiki Komatsuzaki. A bad arm existence checking problem: How to utilize asymmetric problem structure? *Mach. Learn.*, 109(2):327–372, 2020.

[31] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *NIPS*, 24, 2011.

[32] Hideaki Kano, Junya Honda, Kentaro Sakamaki, Kentaro Matsuura, Atsuyoshi Nakamura, and Masashi Sugiyama. Good arm identification via bandit feedback. *Mach. Learn.*, 108:721–745, 2019.

[33] Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In 48^{th} *CISS*, pages 1–6. IEEE, 2014.

[34] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identification in multi-armed bandit models. *JMLR*, 17:1–42, 2016.

[35] Shahin Shahrampour, Mohammad Noshad, and Vahid Tarokh. On sequential elimination algorithms for best-arm identification in multi-armed bandits. *IEEE Trans. Signal Process.*, 65(16):4281–4292, 2017.

[36] Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sébastien Bubeck. Multi-bandit best arm identification. *NIPS*, 24, 2011.

[37] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. *NIPS*, 27, 2014.

[38] Kevin G Jamieson and Lalit Jain. A bandit approach to sequential experimental design with false discovery control. *NIPS*, 31, 2018.

[39] Hampus Gummesson Svensson, Esben Jannik Bjerrum, Christian Tyrchan, Ola Engkvist, and Morteza Haghir Chehreghani. Autonomous drug design with multi-armed bandits. In *Conf. on Big Data*, pages 5584–5592. IEEE, 2022.

[40] Mengyan Zhang, Russell Tsuchida, and Cheng Soon Ong. Gaussian process bandits with aggregated feedback. In *AAAI Conf. on Artificial Intelligence*, volume 36, pages 9074–9081, 2022.

[41] Sattar Vakili, Nacime Bouziani, Sepehr Jalali, Alberto Bernacchia, and Dashan Shiu. Optimal order simple regret for gaussian process bandits. *NIPS*, 34:21202–21215, 2021.

[42] Jacob D Abernethy, Kareem Amin, and Ruihao Zhu. Threshold bandits, with and without censored feedback. *NIPS*, 29, 2016.

[43] James Cheshire, Pierre Ménard, and Alexandra Carpentier. Problem dependent view on structured thresholding bandit problems. In *ICML*, pages 1846–1854. PMLR, 2021.

[44] Koji Tabata, Atsuyoshi Nakumura, and Tamiki Komatsuzaki. Classification bandits: Classification using expected rewards as imperfect discriminators. In *PAKDD workshop on MLMEIN*, pages 57–69. Springer, 2021.

[45] Tatsuya Hayashi, Naoki Ito, Koji Tabata, Atsuyoshi Nakamura, Katsumasa Fujita, Yoshinori Harada, and Tamiki Komatsuzaki. Gaussian process classification bandits. *Pattern Recognit.*, 149:110224, 2024.

[46] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.*, 9(2):513–530, 2018.

[47] Nobuya Tsuji, Pavel Sidorov, Chendan Zhu, Yuuya Nagata, Timur Gimadiev, Alexandre Varnek, and Benjamin List. Predicting highly enantioselective catalysts using tunable fragment descriptors. *Angew. Chem. Int. Ed.*, 62(11):e202218659, 2023.

[48] Aditya Raymond Thawani, Ryan-Rhys Griffiths, Arian Jamasb, Anthony Bourached, Penelope Jones, William McCorkindale, Alexander Aldrick, et al. The photoswitch dataset: a molecular machine learning benchmark for the advancement of synthetic chemistry. *ChemRxiv.*, 2020.

[49] Ryan-Rhys Griffiths, Jake L Greenfield, Aditya R Thawani, Arian R Jamasb, Henry B Moss, Anthony Bourached, Penelope Jones, William McCorkindale, Alexander A Aldrick, Matthew J Fuchter, et al. Data-driven discovery of molecular photoswitches with multioutput gaussian processes. *Chem. Sci.*, 13(45):13541–13551, 2022.

[50] Philippe Rigollet and Jan-Christian Hütter. High-dimensional statistics. *arXiv preprint arXiv:2310.19244*, 2023.

[51] Neel S Madhukar, Prashant K Khade, Linda Huang, Kaitlyn Gayvert, Giuseppe Galletti, Martin Stogniew, Joshua E Allen, Paraskevi Giannakakou,

and Olivier Elemento. A bayesian machine learning approach for drug target identification using diverse data types. *Nat. Commun.*, 10(1):5221, 2019.

[52] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[53] Alkis Gotovos. Active learning for level set estimation. Master's thesis, Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2013.

[54] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv:0912.3995*, 2009.

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

[56] Ramil I Nugmanov, Ravil N Mukhametgaleev, Tagir Akhmetshin, Timur R Gimadiev, Valentina A Afonina, Timur I Madzhidov, and Alexandre Varnek. Cgrtools: Python library for molecule, reaction, and condensed graph of reaction processing. *J. Chem. Inf. Model.*, 59(6):2516–2521, 2019.

[57] Shifa Zhong and Xiaohong Guan. Count-based morgan fingerprint: A more efficient and interpretable molecular representation in developing machine learning-based predictive regression models for water contaminants' activities and properties. *Environ. Sci. Technol.*, 57(46):18193–18202, 2023.

[58] Alexandre Varnek, Denis Fourches, Frank Hoonakker, and Vitaly P Solov'ev. Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures. *J. Comput.-Aided Mol. Des.*, 19:693–703, 2005.

[59] Igor Baskin and Alexandre Varnek. Fragment Descriptors in SAR/QSAR/QSPR Studies, Molecular Similarity Analysis and in Virtual Screening. In *Chemoinformatics Approaches to Virtual Screening*. The Royal Society of Chemistry, 09 2008.

[60] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC press, 2015.

[61] Greg Landrum. Rdkit: Open-source cheminformatics software. 2016.

[62] Harry L Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *J. Chem. Doc.*, 5(2):107–113, 1965.

[63] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *J. Chem. Inf. Model.*, 50(5):742–754, 2010.

The End