Doctoral Thesis

博士論文

Molecule Identification from NMR Spectra with Machine Learning

(機械学習を用いたNMRスペクトルからの分子同定)

Zhang Jinzhe

張　金哲

ACKNOWLEDGMENTS

On the very outset of this dissertation, I would like to extend my sincere and heartfelt obligation towards all the personages who have helped and encouraged me in this endeavor.

First of all, I would like to express my gratitude to Prof.Koji Tsuda for his guidance, his patience and his careness towards all of his students. He is being truly resourceful, supportive and understanding during the course of my PhD study. What I have learned from him, in addition to the scientific knowledge and thinking process, is his mild but insistent spirit in extending the boarder of our understanding.

I extend my gratitude to Prof.Jun Kikuchi for providing meaningful insights about NMR spectroscopy. Together with my co-advisors Prof. Kiyoshi Asai and Prof. Hisanori Kiryu and referee of my thesis committee Prof.Ryo Tamura, they have provided fruitful comments and firm encouragements.

I would also thank Prof.Kei Terayama for extensively guided me during the first year of my PhD study. He spent tremendous energy in teaching me high performance computing system and showing me the level of thoroughness one research requires.

I thank Dr.Masato Sumita for being an chemistry expert, for being a friend and for tolerating my harassment using random chemistry questions.

My thank also goes to Mrs.Xiaolin Sun, Dr.Koki Kitai, Mrs.Jiawen Li, Mr.Naruki Yoshikawa, Mr.Zihao Yang, Mrs.Duolin Li for being fully supportive.

I acknowledge my gratitude to my parents, Mrs.Xinxia Hu and Mr.Fuxiang Zhang for achieving a 29-year long dedication in loving, supporting and guiding me.

A special thank to Mrs.Yixuan Guo for loving and enduring the best and the worst of me.

It is with this uniquely supportive environment that you all collectively created made this dissertation possible.

ABSTRACT

Structure elucidation is the process of determining the structure of a chemical compound from a sample. It is a basic but essential task in biology and chemistry research. For organic compounds, structure elucidation is often involved with Nuclear Magnetic Resonance (NMR) spectroscopy. NMR spectroscopy is a technique that observes local magnetic fields around atomic nuclei. NMR spectroscopy machine excites the sample with a radio frequency pulse, a nuclear magnetic resonance response is then obtained. This nuclear magnetic resonance response, also known as a free induction delay (FID), can be converted into a spectrum with a Fourier Transformation.

On the NMR spectrum, functional groups in a molecule appear as peaks which characterizes themselves depending on their chemical environment in the molecule. Chemists can often readout the knowledge behind the chemical shifts where the peaks of some groups appear and assembly information together to obtain structural information of the sample molecule. Therefore, the approximate chemical structure of the target molecule can be confirmed from the NMR spectrum. In addition to structure elucidation by chemists, it is also possible to match observed NMR spectrum with existing NMR spectrum in the database. Although many previously observed NMR spectra are accumulated in public databases, they cover only a tiny fraction of the chemical space.

To overcome the limitation of current structure elucidation methods with NMR spectroscopy, an automated structure elucidation method independent to database content is desired. Recent progress in machine learning has enabled the development of de novo molecule generators which are expected to design molecules with desired properties. Previously, our lab developed a molecule generator, ChemTS, which combines Monte Carlo tree search (MCTS) with a recurrent neural network (RNN), and successfully showed that ChemTS coupled with quantum chemical calculations can produce realistic molecules that have desired properties. So far, most de novo molecule generators have only been tested or applied on quantifiable chemical properties such as gaps between the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO). As 1H NMR spectra are highly characteristic of individual compounds, we consider 1H NMR spectra as one of its molecular properties.

In this thesis, we propose NMR-TS, a machine-learning-based python library, to automatically identify a molecule from its NMR spectrum. NMR-TS discovers candidate molecules whose NMR spectra match the target spectrum by using deep learning and density functional theory (DFT)-computed spectra.

As a proof-of-concept, we identify prototypical metabolites from their computed spectra. After an average 5451 DFT runs for each spectrum, six of the nine molecules are identified correctly, and proximal molecules are obtained in the other cases. This encouraging result implies that de novo molecule generation can contribute to the fully automated identification of chemical structures.

In addition, we will discuss about another main obstacle of automatic structure elucidation from NMR spectrum which is the distance metric between two NMR spectra. Theoretically, measuring/predicting same molecular structure should result exactly the same NMR spectra using different machine or predictor. However, this is not true in reality. As a sophisticated measurement, even a slight error could cause mis-classification between NMR obtaining from different sources. In this thesis, we summarize the current distance metrics and suggest potential new distance metric by training a neural network as a customized distance metric.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Our understanding about the nature expands with our observation capability. The average of human eyes can distinguish objects as small as about 0.1 mm. This remains to be our observation limit until the invention of optical microscope in 1590. Using optical microscope, Robert Hooke and Antonie van Leeuwenhoek both independently achieved to observe cellular structure which leads to the creation of microbiology. With continuous improvement on optical microscope, soon our limit of observation become the wavelength of visible light. If the size of the object we are observing is smaller than the wavelength of visible light, it is hopeless to use visible light as an observation tool. In 1930s, Isidor Rabi discovered the interaction between the nuclei magnetic moment and the magnetic field, and first described the phenomena of Nuclear Magnetic Resonance (NMR) . This work is then soon expanded by Felix Bloch and Edward Purcell in 1945 to apply on liquids and solids [1] and both Purcell group and Bloch group developed NMR spectroscopy independently in 1940s and 1950s. NMR spectroscopy is a spectroscopic technique to obtain structural information of chemical compounds by observing local magnetic fields around atomic nuclei. As the intramolecular magnetic field around an atom would make an impact on the resonance frequency and will reveal details of the electronic structure of a molecule and its individual functional groups. NMR spectroscopy expands again our ability to observe and provides detailed information about the structure, dynamics and the chemical environment of a molecule. Being able to observe and identify this information is a critical step for our understanding about mechanisms at an atomic or molecular level.

1.1 Current state of structure elucidation from NMR spectrum

Structure elucidation is the process of determining the structure of a chemical compound from a sample. It is a basic but essential task in biology and chemistry research. Under the context of NMR spectroscopy meaning the determination of a chemical structure an unknown NMR spectrum [2]. Currently, there are mainly two ways of structure elucidation. The first way is if the spectrum of the observed sample fully coincides with a reference spectrum, usually stored in an NMR database, it implies the observed sample is identical to the reference spectrum. We can therefore retrieve the chemical structure from the reference database.[3, 4, 5] The second way is to manually induct the structure behind the spectrum using pieces of information obtained from the NMR spectrum. This is usually the most complicated task.

The drawback of both current approaches are obvious. In the approach of retrieving structural information from a database of NMR spectrum, the feasibility of such approach is highly dependent on prior database knowledge. If the molecule structure in question is previously unknown or simply the information is not stored in the database we are searching, such approach will most certainly fail. In the second approach of manually induct the structure from NMR spectrum, it is highly dependent on both human involvement and expertise of chemists. This demand of human effort could be an inhibitory factor for massive or repetitive elucidations.

1.2 Automated robotic laboratory

Recently, automated robotic laboratory systems have received considerable attention for high-throughput material design. The automated system allows machine learning algorithm to require and perform chemical experiments, measure the result of each experiment, and the algorithm will then decide what to do next. Studies have demonstrated that automated robotic laboratories can perform chemical reactivity tests under different reaction conditions guided by machine learning algorithm [6, 7, 8]. One of the key steps in this automated system is the measurement step. We need this measurement step as a feedback from actual experiment and feed the information to the algorithm. For instance, in the experiment of improving yield of reaction product, this measurement should be the amount of desired product after reaction. In fact, a bench-top NMR spectroscopy is the exact technique used to do this quantitative measurement [8]. If we can perform not only quantitative measurement, but also qualitative measurement, it will enable the automated system for a wider range of application such as automated discovery of retro-synthetic pathway. For now, although the acquisition of NMR spectrum can be fully automated, the interpretation from NMR spectrum to a chemical structure can be fully automated especially when it involves chemical structure which is not in the database. An automated structure elucidation process is the one piece of puzzle missing here.

1.3 Metabolites ”dark matter”

Also, study shows that, on a biological sample that we extracted from marine organism, there are about 98 percents of peaks on the spectrum that does not belong to any structure that we have in our knowledge. We can match only about 2 percents of peaks to a known chemical structure. This circumstances is referred as ”dark matter” in [9]. The existence of biological ”dark matter” gives us a reason to create a method that elucidate the chemical structure from a spectrum without database knowledge. Although an automated structure will be the remedy for every problem involved (for example, one of hardest problem for biological extraction spectrum is the interaction between different chemical structures), it would be one step forward towards the understanding of the metabolites dart matter.

1.4 An automated system for structure elucidation from NMR spectrum

Recent progress in machine learning has enabled the development of de novo molecule generators [10, 11, 12, 13, 14, 15, 16, 17]. *de novo* molecule generator is a type of algorithms that will automatically design a chemical structure with desired chemical properties, this is also known as inverse molecular design [11]. Usually, molecule generator takes a desired property(or a set of properties for multi-objective design) as input and generate a set of chemical compound candidates which should potentially has close-to-desired property as output. For instance, our laboratory has previously developed a molecule generator namely ChemTS [14], which combines Monte Carlo tree search (MCTS) with a recurrent neural network (RNN). We have successfully showed that ChemTS can be coupled with quantum chemical calculations and produce realistic molecules that have a certain desired properties [18]. However, to date, most *de novo* molecule generators have only been designed or tested on quantifiable chemical properties such as HOMO-LUMO gaps (gaps between HOMO, the highest occupied molecular orbital, and LUMO, the lowest unoccupied molecular orbital). As ^{1}H NMR spectra are highly characteristic for each individual chemical structure, we can consider ^{1}H NMR spectra as one of its molecular properties. The approach that we want to take is to ask a molecule generator to design a chemical compound that has a NMR spectrum as close as possible to the measured experimental NMR spectrum. If the molecule generator can generate a molecule that has ”exactly the same” NMR spectrum compare to the measured one, we can then consider that we recovered the chemical structure behind the measured NMR spectrum.

During my PhD study, We developed a python library named NMR-TS to identify a known or unknown molecule from its NMR spectrum by designing molecules that have as similar as possible ^{1}H NMR spectra compare to the input spectrum. As a proof-of-concept work, we implemented this concept and evaluated its performance on the ^{1}H NMR spectra of nine molecules. All nine molecules were not included in the dataset that NMR-TS has encountered. NMR-TS succeeded in correctly identifying six of the nine molecules from their ^{1}H NMR spectra, whereas proximal molecules were obtained in the other three cases.

In addition, we looked into the similarity metric between two NMR spectra. Each molecule has a unique NMR spectrum, in a lot of the cases, for two chemical compound with similar chemical structures, their NMR spectra tends to be similar. However, this is not always true. In some cases, NMR spectrum of some largely different chemical structures could look similar to each other. Also, current NMR spectrum prediction methods, based either on computational chemistry[19] or machine learning [20, 21, 22, 23, 24, 25, 26] are accurate, but not accurate enough for information retrieval tasks when the amount of NMR spectra in the database is massive. For this purpose, we tried to explore different strategy to tackle the distance metric between two NMR spectra in order to better answer questions like: 1) Are these NMR spectra represents the same molecule? 2) If not, how different are they?

CHAPTER 2

NMR SPECTROSCOPY

2.1 Physical Basics of Nuclear Magnetic Resonance

2.1.1 Spin

Spin is an intrinsic form of angular momentum carried by elementary particles, composite particles and atomic nuclei [27]. Nuclear magnetic resonance is mainly caused by the spin of atomic nuclei. Different atomic nuclei have different type of spin. They can be represented by the spin quantum number I of the nucleus. There is a certain relationship between the spin quantum number, the mass number and atomic number of the atom, which can be roughly divided into three categories, as shown in the following table.

Type	**Mass Number**	**Atomic Number**	**Spin Quantum Number**	**NMR**
I	Even	Even	0	No
II	Even	Odd	1, 2, 3, ...	Yes
III	Odd	Odd or Even	1/2, 3/2, 5/2, ...	Yes

2.1.2 Nuclear Magnetic Resonance

Nuclear Magnetic Resonance describe a physical phenomenon of nuclei in a strong constant magnetic field being perturbed by another oscillating magnetic field. After the perturbation, nuclear spin will try to restore to a direction that is aligned to the strong constant magnetic field. This restoration process will produce an electromagnetic signal with a frequency. This frequency of electromagnetic signal will decay in time until the oscillation stops. The variation range of the oscillation frequency is near resonance to the intrinsic frequency of the nuclei, which mainly depends on the chemical environment and the magnetic prop reties of the isotope involved. In the context of a molecule or a protein complex, as each nuclei is submitted to a unique chemical environment the resonance frequency of each nuclei should be unique. As each nuclei has a unique frequency, the combination of all unique frequency which is represented in form of an oscillation should also be unique for each distinct chemical structure.

For any nuclei that could interact with the magnetic field, the nucleus must have an intrinsic nuclear magnetic moment and angular momentum. The later condition is met when the nuclei has a non-zero nuclear spin, which in most of the case represents that the nuclei has an odd number protons and/or an odd number of neutrons.

^{1}H and ^{13}C and two types of commonly used nuclei in NMR spectroscopy.

2.2 Nuclear Magnetic Resonance Spectroscopy

Nuclear magnetic resonance spectroscopy, commonly known as NMR spectroscopy, is a spectroscopic technique to obtain structural information of chemical compounds by observing local magnetic fields around atomic nuclei. The substance in question will be placed in the center in a strong and constant magnetic field using a tool called NMR probe. The constant magnetic field is often achieved by using homogeneous permanent magnet for low-frequency NMR spectroscopy or superconducting magnet for high-frequency NMR spectroscopy. Once the substance is placed in the NMR probe at the designated position inside of a NMR spectroscopy machine, as described in the previous section, the magnetic field will make the spin of NMR active atoms inside of the sample align to the constant magnetic field.

In the second step, we need to add a perturbation to the aligned nuclear spins by adding a weak oscillating magnetic field. Usually this oscillating magnetic field is referred to a radio-frequency pulse. The NMR probe contains a radio frequency coils which is used to excite the sample by applying this radio-frequency pulse and record the responses. The pulse emits an excitation of the nuclei sample with radio waves into nuclear magnetic resonance in order to produce the NMR signal.

Upon excitation of the sample, a nuclear magnetic resonance response, also called a raw time-domain Free Induction Decay (FID), is detected by sensitive radio receivers. Each single FID captured could have a low signal-to-noise ratio, but this can be improved by doing multiple excitations and averaging FID records. A Fourier Transformation is then carried out to transform time-domain FID into frequency-domain spectrum which is unique or highly characteristic to individual compounds.

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} \mathrm{d}x \tag{2.1}$$

An example of the frequency-domain spectrum obtained after Fourier Transformation is the following:

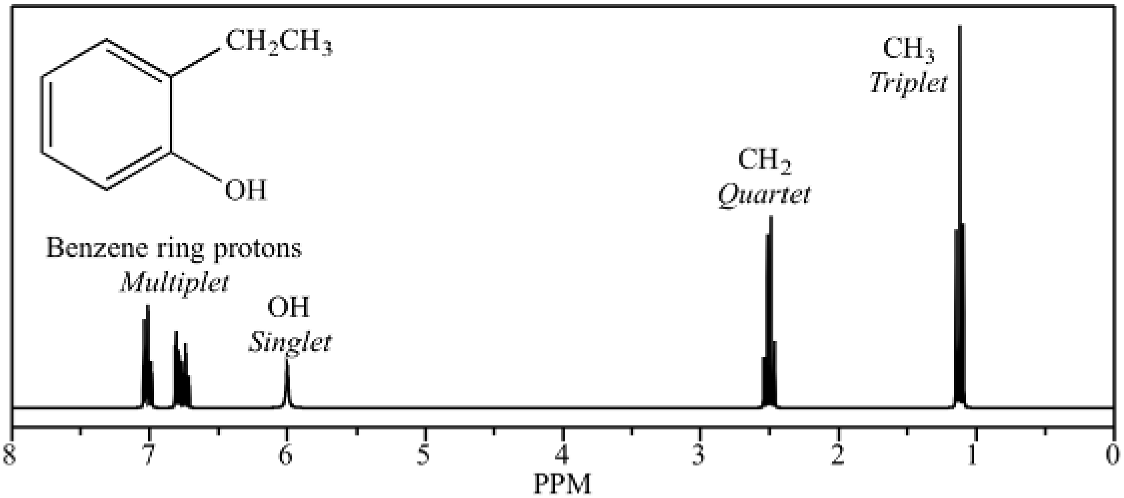


Figure 2.1: ^{1}H NMR Spectrum of 2-ethyl-phenol. [28]

On the NMR spectrum, each peak represents a resonant frequency in the sample, also known as chemical shift. The position and the number of chemical shift are key information for the structure elucidation.

Unless explicitly noted otherwise, all chemical shift in this thesis are expressed in parts per million (ppm) with

$$\delta = \frac{v_{sample} - v_{ref}}{v_{ref}} \tag{2.2}$$

where v_{sample} is the absolute resonant frequency of a chemical shift and v_{ref} is the absolute resonant frequency of a reference compound. In our study and very commonly in other studies, Tetramethylsilane (TMS) is used as the reference compound.

2.3 Structure elucidation from NMR spectrum

Structure elucidation is the task to map or deduct from the NMR spectrum to a chemical structure. Usually, for any NMR spectrum obtained, we first search the spectrum or the position of each chemical shift in a NMR spectrum database. Depends on the outcome of this search, the elucidation goes into two direction. If the NMR spectrum is known in the database, which means the spectrum match closely to a known structure in the database, scientists can directly readout the chemical structure. Otherwise, the scientist will need to deduct the chemical structure based on the information given from the spectrum.

For different type of NMR spectroscopy, the structure elucidation procedures are not the same. Here we show the example with a ^{1}H NMR spectrum, which is a fairly commonly used type of NMR spectroscopy.

To deduct the chemical structure from the NMR spectroscopy, the first step is usually assign each peak to a particular functional group. As mentioned earlier, the chemical shift of a nuclei depends on its chemical environment. In organic chemical compounds, nuclei of same functional group tends to has a similar frequency despite the fact that their chemical environment can be impacted by the rest part of the molecules. As a result, we can usually assign a functional group to a particular peak on the NMR spectrum depends on its chemical shift and its shape.

Once we assigned each peak to a functional group. The shape of each peak, also known

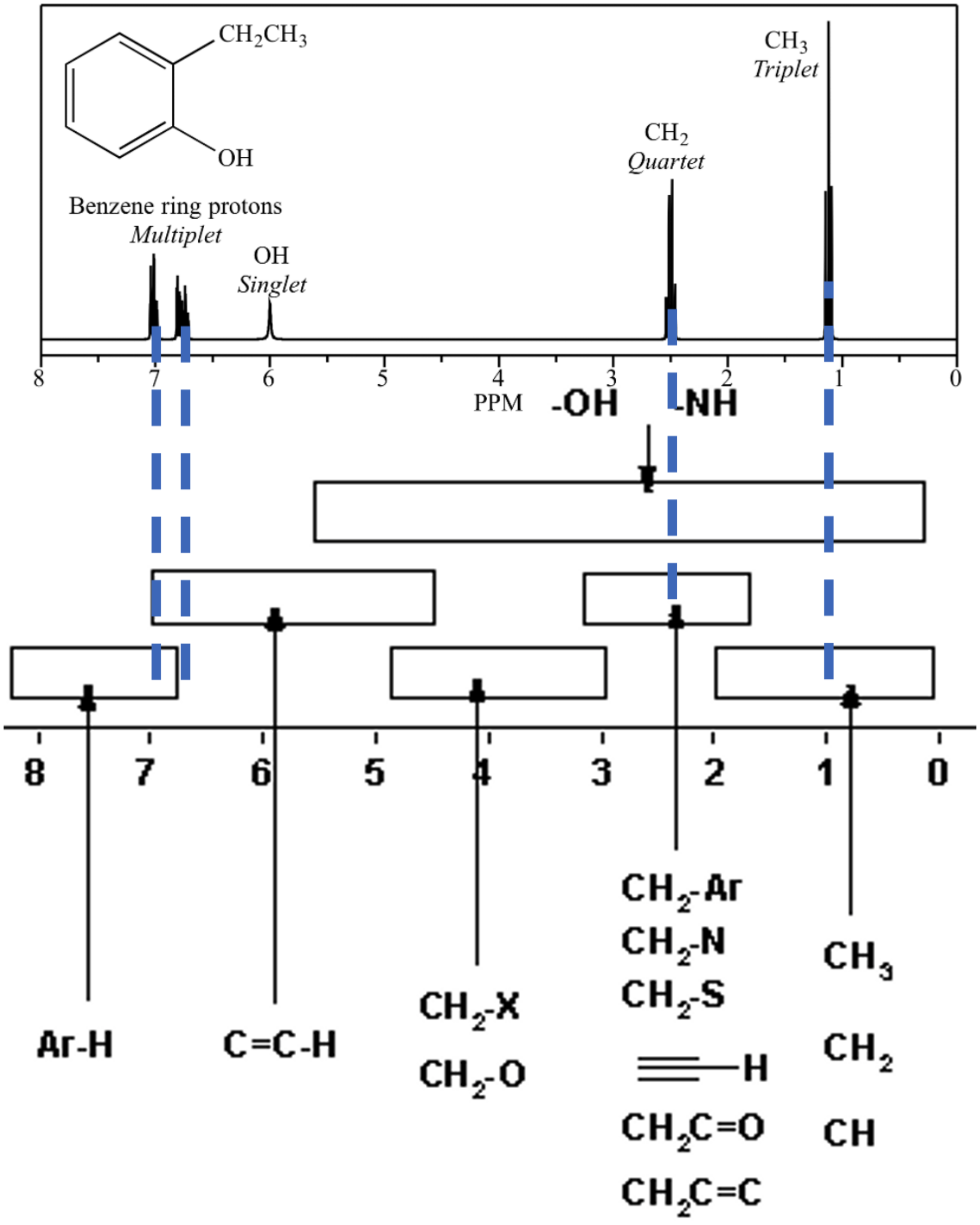


Figure 2.2: Peak assignment

as the J-coupling effect, would be able to show the number of Hydrogen that neighbouring function group(s) contain(s). If we consider each peak on the spectra as a piece of Jigsaw puzzle, then peak assignment will reveal the content printed on the piece of puzzle, and the J-coupling effect will reveal the shape of each puzzle which directly decides how this piece of puzzle can concatenate with the other pieces.

The most challenging job of chemists, is to solve this Jigsaw puzzle game by putting different functional group into reasonable position and reveal the whole molecule structure.

CHAPTER 3
DE NOVO MOLECULE GENERATORS

For many fields ranging from aerospace vehicle manufacturing to pharmaceutical, the ability to find and use a material or a chemical compound that fits to the demand could be the decisive factor for the success of the project. For example, for many of the diseases, we understand deeply about the mechanism that caused the disease, but we could not find the right chemical compound that prevent the mechanism and don't cause serious unwanted effect. This could be a ligand that bands tight enough to a protein binding site, or a inhibitor compound that prevent β-amyloid precursor protein from accumulating inside of brain cell.

Traditionally in chemistry, finding and designing a chemical structure with a certain property is mainly achieved by manually designed chemical structure from chemistry experts based on their previous experiences and chemistry insights. However, the relationship between chemical structure and a certain property is usually highly complicated and abstracted. Human knowledge and insights are often not enough to make an accurate deduction.

In recent years, the task of designing a chemical compound matches a desired properties is more and more studied as a black-box optimization problem. In this approach, we imagine that all theoretically possible chemical compounds as a high dimensional space, often referred as chemical space, the task of finding a chemical compound that has a special property is then converted into a task of black-box optimization in a huge space. The black-box represents the relationship between a chemical structure (a point in the chemical space) to a property value.

Computer-aided molecular design was initially tackled by combining predefined chemical fragments[29]. Ikebata et al.[30] then achieved to design molecules in a special form of representation namely SMILES. Generating SMILES[31] form of molecules meaning that the molecule generation task can be converted into a linguistic or sequence generation task. Gomez-Bombarelli et al.[32] were the first to apply variational autoencoder(VAE) to the molecule generation task. Kusner et al.[33] then enhanced the VAE generator to employ grammatical information and created grammatical(GVAE). In 2016, Segler et al.[34] showed that a recurrent neural network(RNN) can be trained to generate SMILES with high validity. Yang et al. followed up on this idea and created ChemTS[14] which combines RNN and Monte Carlo Tree Search (MCTS) to gradually generate SMILES form of molecules towards a desired property(or scoring function).

In our study, we choose to use ChemTS as our generator for two reasons: 1) ChemTS can be parallelized to make the best use of high performance computing system, 2) the concept of combining ChemTS with quantum chemistry calculation is already proven in a previous study[18].

3.1 ChemTS

ChemTS is an efficient Python library of de novo molecular design. ChemTS takes a database of SMILES [31] strings and an evaluation function, which evaluates matching level of the generated molecule to the desired property. Starting from a root node that represents the beginning of a SMILES string, the ChemTS algorithm builds a monte carlo search tree. On each level of the monte carlo search tree, each node represents a possible SMILES character at this position. Similar to traditional MCTS, ChemTS consists four steps: 1) selection, 2) expansion, 3) simulation and 4) backpropagation, details of each steps are given in the original paper [14].

In the selection step, the algorithm traverse the search tree starting from the root until reaching a leaf, each time arriving to a new node, the algorithm recursively choosing one child node that has the maximum upper confidence bound (UCB) based score at each branch. The path from the root to the leaf node is constituted as a series of characters, this character sequences is then used as the prefix of the SMILES.

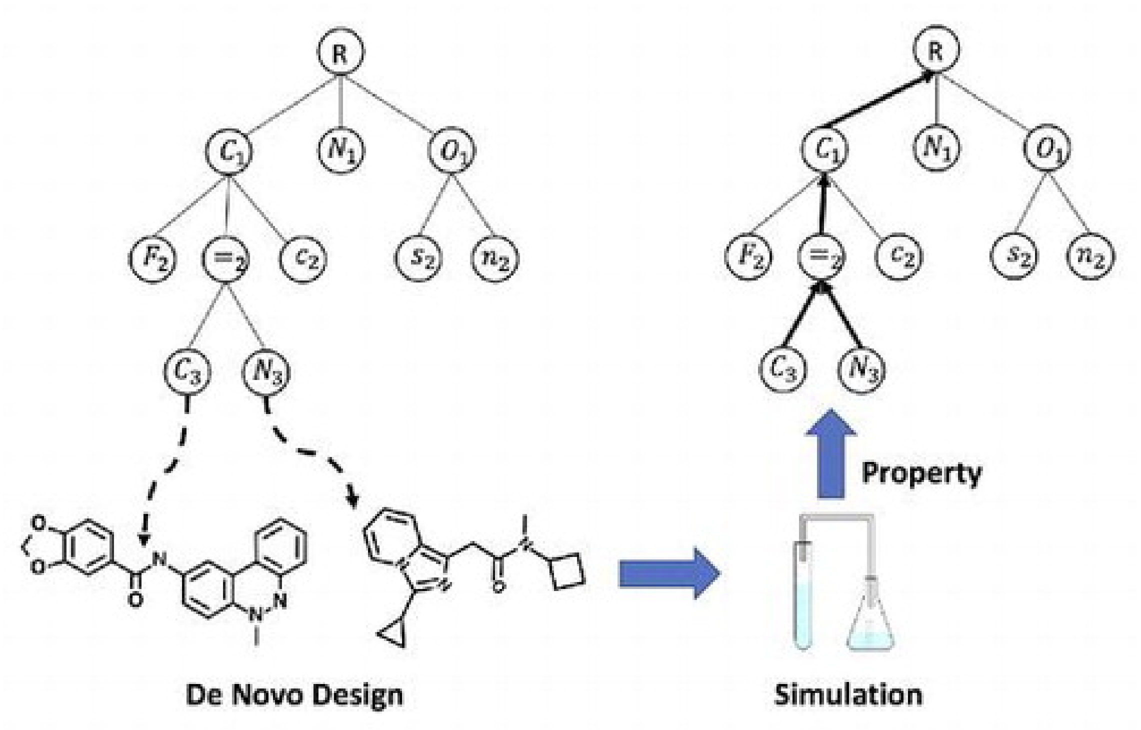


Figure 3.1: Concept of ChemTS [14]

In the expansion step, all or several possible (depends on expansion strategy) candidate characters are added as child node of a current leaf node. Upon tree expansion, the traverse from root node to the leaf node will constitute a string which serves as the prefix input for the RNN. Given the prefix as input, RNN can repeatedly predict the next symbol based on prefix up to this point. This process will keep repeating until the prediction of RNN to be a terminal symbol meaning that a complete SMILES string is generated[34]. We will then evaluate the generated SMILES using the evaluation function. The obtained score will then traverse backwards to the root node, and update all traversed node alongside. This final step is called backpropagation.

The RNN of ChemTS is supposed to be trained on an input molecules database in which molecules are represented in form of SMILES. Ideally, the molecular characteristic of the training set should match as closely as possible of the expected candidate molecule. For example, if we don't expect to have any molecule with positive or negative charge, then the training set of the molecules shall not contain any charged molecules.

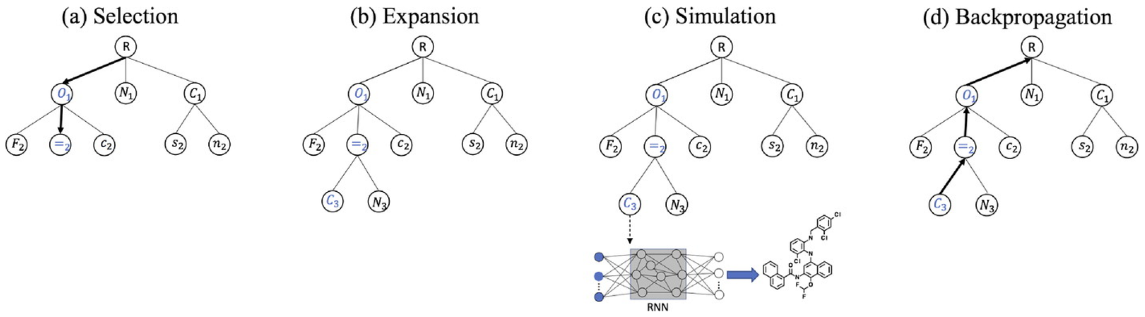


Figure 3.2: Illustration of four steps of Monte Carlo Tree Search [14]

3.2 Parallel ChemTS with virtual loss

In many cases for molecular design, the evaluation function involves in using quantum chemistry computation such as DFT. A single DFT run could take somewhere between minutes to days of computation time. The stand-alone version of ChemTS proceed evaluation in a iterative way which is not optimal when evaluation takes significant amount of time. Therefore, we wish to make a parallelization to accelerate the process.

Currently there are several strategy to parallel monte carlo tree search. Cazenave et al. [35] has proposed leaf parallel and root parallel strategy for MCTS in 2007. Later, Chaslot et al. [36] proposed another parallel strategy using virtual loss. Both approaches are also suggested for ChemTS in [37]. In our study, we employed the virtual loss strategy to parallelize MCTS using OpenMPI. It is worth mentioning that after the publication of NMR-TS, a newer parallel version ChemTS is published using hashing [38]. The hashing strategy has theoretically a better performance.

Based on our strategy with virtual loss, the UCB scoring defined as the following:

$$UCB_i = \frac{s_i}{v_i + w_i} + CP_i \frac{\sqrt{v_p + w_p}}{1 + v_i + w_i} \tag{3.1}$$

with:

s_i: the total score obtained by node i.

w_i: the total visit number of i.

v_p: the total virtual visit number of i (virtual loss).

v_p: the total visit number of parent node p of i.

v_i: the total virtual visit number of i.

p_i: the probability of i among the children of p.

C: a constant that controls the exploration–exploitation trade-off.

CHAPTER 4
NMR-TS

Using a similar concept of ChemTS, we designed NMR-TS, a system that can automatically elucidate the chemical structure from an input NMR spectrum. The system takes two input: 1) an input NMR spectrum, 2) a SMILES dataset as training set of RNN. The goal is to find out what is the chemical structure behind this spectrum.

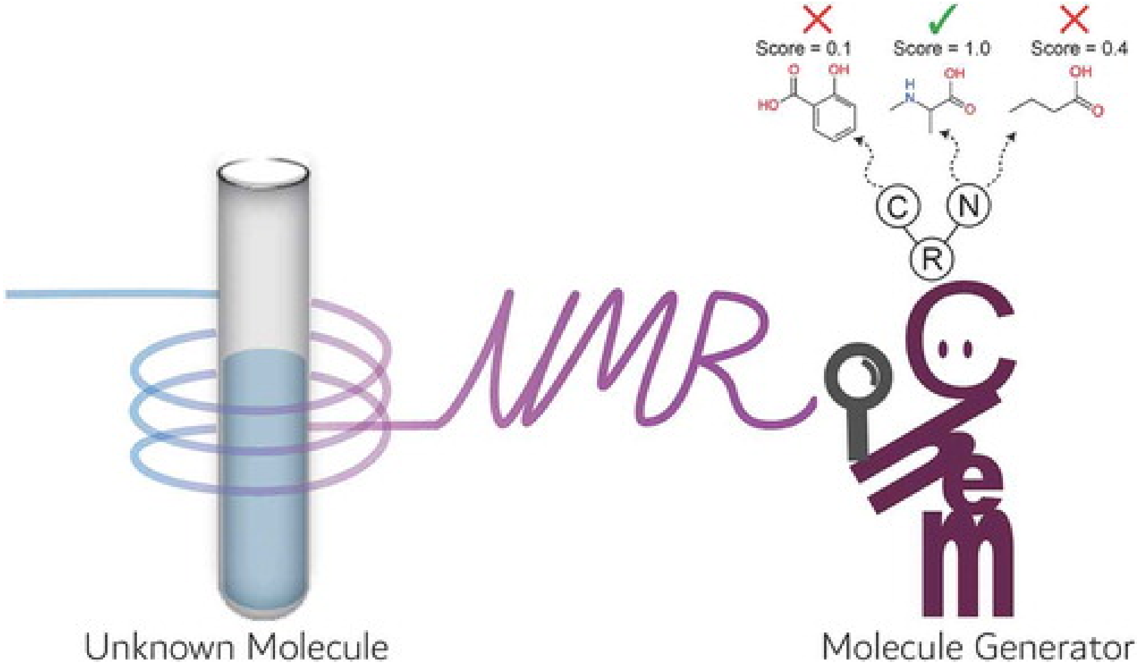
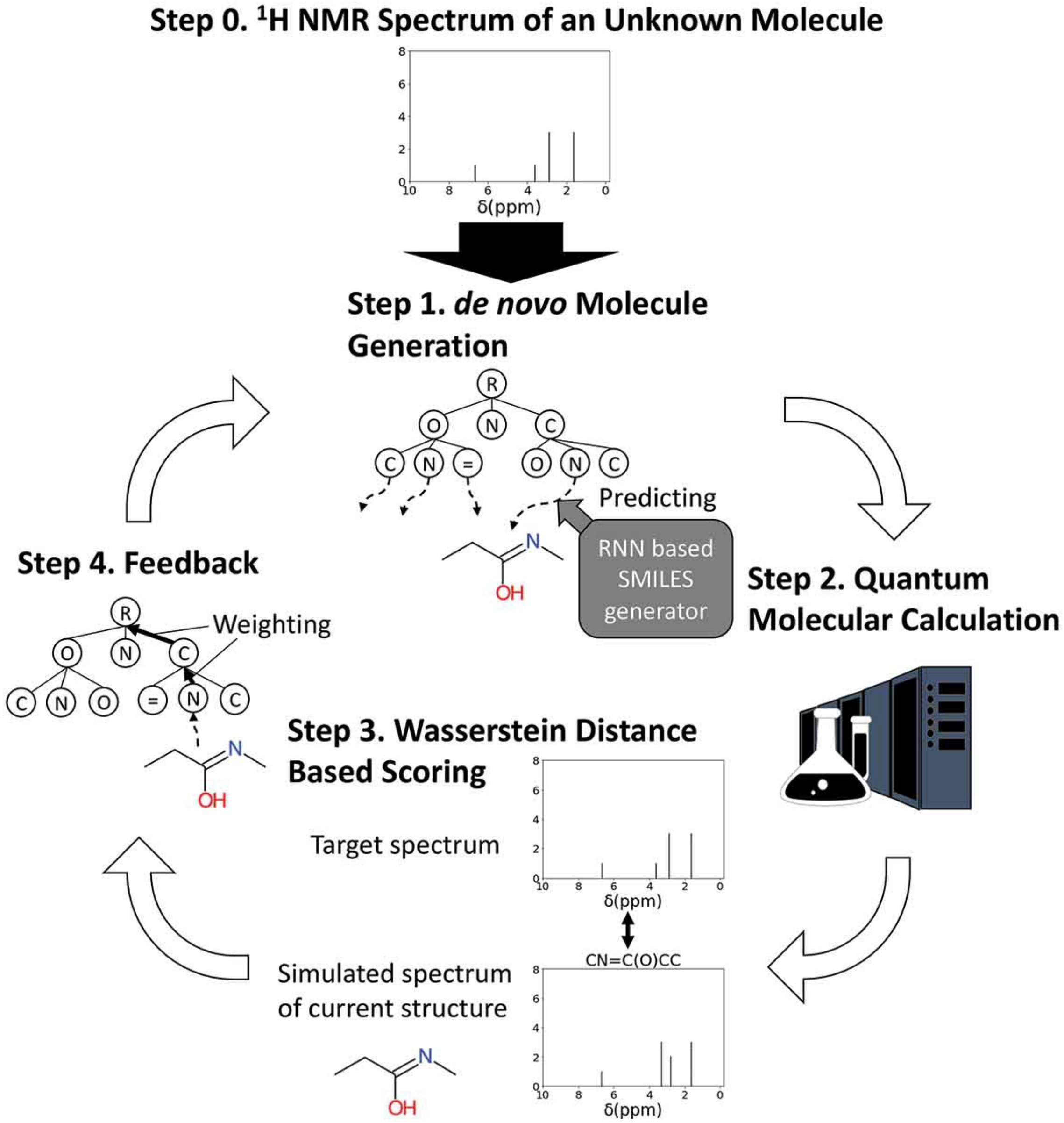


Figure 4.1: NMR-TS

Figure 4.2 illustrate the concept and the workflow of NMR-TS. NMR-TS takes a target NMR spectrum as input (Step 0). Step 1 is the generation step. In this step, we ask our de novo generator to generate a molecule. In step 2, we use DFT to simulate the NMR spectrum of this generated molecule. Once we obtained the NMR spectrum of the generated molecule, we compare the NMR spectrum of this generated molecule to the input NMR spectrum. A score will be obtained about the similarity level between the input spectrum and the simulated spectrum (Step 3), and we will feedback the score to the monte carlo search tree (Step 4). The generator will try to learn from its experiences and make a better guess by generating a molecule which has more similar NMR spectrum comparing to the input NMR spectrum. If we get lucky and obtained a molecule which generates exactly the same spectrum as the input spectrum, it means we successfully elucidated the chemical structure of the input spectrum.



Step 0. ¹H NMR Spectrum of an Unknown Molecule
δ(ppm)
Step 1. de novo Molecule Generation
R
O
N
C
=
Predicting
RNN based SMILES generator
N
OH
Step 2. Quantum Molecular Calculation
Step 3. Wasserstein Distance Based Scoring
Target spectrum
CN=C(O)CC
Simulated spectrum of current structure
Step 4. Feedback
Weighting

Figure 4.2: Concept of NMR-TS

4.1 Input data

4.1.1 Input NMR spectrum

The input NMR spectrum is assumed to be the experimental measured NMR spectrum that we need to elucidate its chemical structure. This is also referenced as the target spectrum in the rest of this thesis because it is this NMR spectrum that our molecule generator will try to recreate.

4.1.2 SMILES dataset

The SMILES dataset is used to train the RNN to generate complete and valid SMILES string. Research shows that the RNN tends to generate similar structure (SMILES string) than its training set and it is possible to make ”focused generation” based on our prior knowledge about the potential characteristics of the candidate molecule [34].

4.2 NMR spectrum prediction

In each iteration, NMR-TS need to simulate the NMR spectrum of generated molecule in order to score it. In order to simulate the 1H NMR spectrum of the generated molecule, we first convert the RNN generated SMILES into canonical SMILES. Then we convert the canonical SMILES into a 3D structure using RDkit library with the random seed fixed to 1. The canonization and the fixation of random seed is for the purpose of fix conformer generated. If we generated the same chemical structure but converted into a different conformer for the reason of random seed, their NMR spectrum might be different and our algorithm would not be able to spot the perfect match is found. After converted into 3D molecular structure, we compute the NMR spectrum using density functional theory (DFT) [39] at the B3LYP/3-21G* level on the optimized structure at the universal force field (UFF) level. We used gauge-invariant atomic orbital(GIAO) method to compute the magnetic shielding tensors. All chemical shifts are standardized to the reference molecule of TMS. DFT calculations are performed using Gaussian 16 package [40]

4.3 Wasserstein distance based evaluation function

Wasserstein distance [41], also known as the Kantorovich-Rubinstein metric or the earth mover's distance [42], is a function that describes distance between two distributions. A visual description for Wasserstein distance is to consider two distribution as two pile of sand, if the amount of work equals to the sum of total distance moved for all grains of sand, then the Wasserstein distance is the minimum amount of work to reshape one pile into the other pile's shape.

$$W_p(\mu, v) := (inf_{\gamma \in \Gamma(\mu,v)} \int_{MxM} d(x,y)^p \mathbf{d}\gamma(x,y))^{1/p} \tag{4.1}$$

For the purpose of adapting our evaluation function to MCTS, we added a penalty term. This penalty term takes the number of hydrogen and carbon atoms in the target molecules to constrain the size of molecules generated by NMR-TS. If the number of Hydrogen or Carbon in the generated molecule is different from the target molecule, the overall score will be penalised. This usage of information from the test molecule can be justified as most of exercise of structure elucidation from NMR spectrum will provide molecular formula (such as C_2H_2) and the essential part of the question is to obtain the structural formula.

We defined the evaluation function, namely Wasserstein Score (WS), $Score(M_g, M_t)$ between generated molecule M_g and target molecule M_t:

$$Score(Mg, Mt) = 1 - (WD(M_g, M_t)) + \alpha Penalty(M_g, M_t)) \tag{4.2}$$

where

$$Penalty(M_g, M_t) = |C(M_g) - C(M_t)| + |H(M_g) - H(M_t)| \tag{4.3}$$

with $WD(M_g, M_t)$: the Wasserstein distance between spectrum M_g and spectrum

M_g: the generated molecule

M_t: the target molecule

$C(M)$: number of carbon in molecule M

$H(M)$: number of hydrogen in molecule M

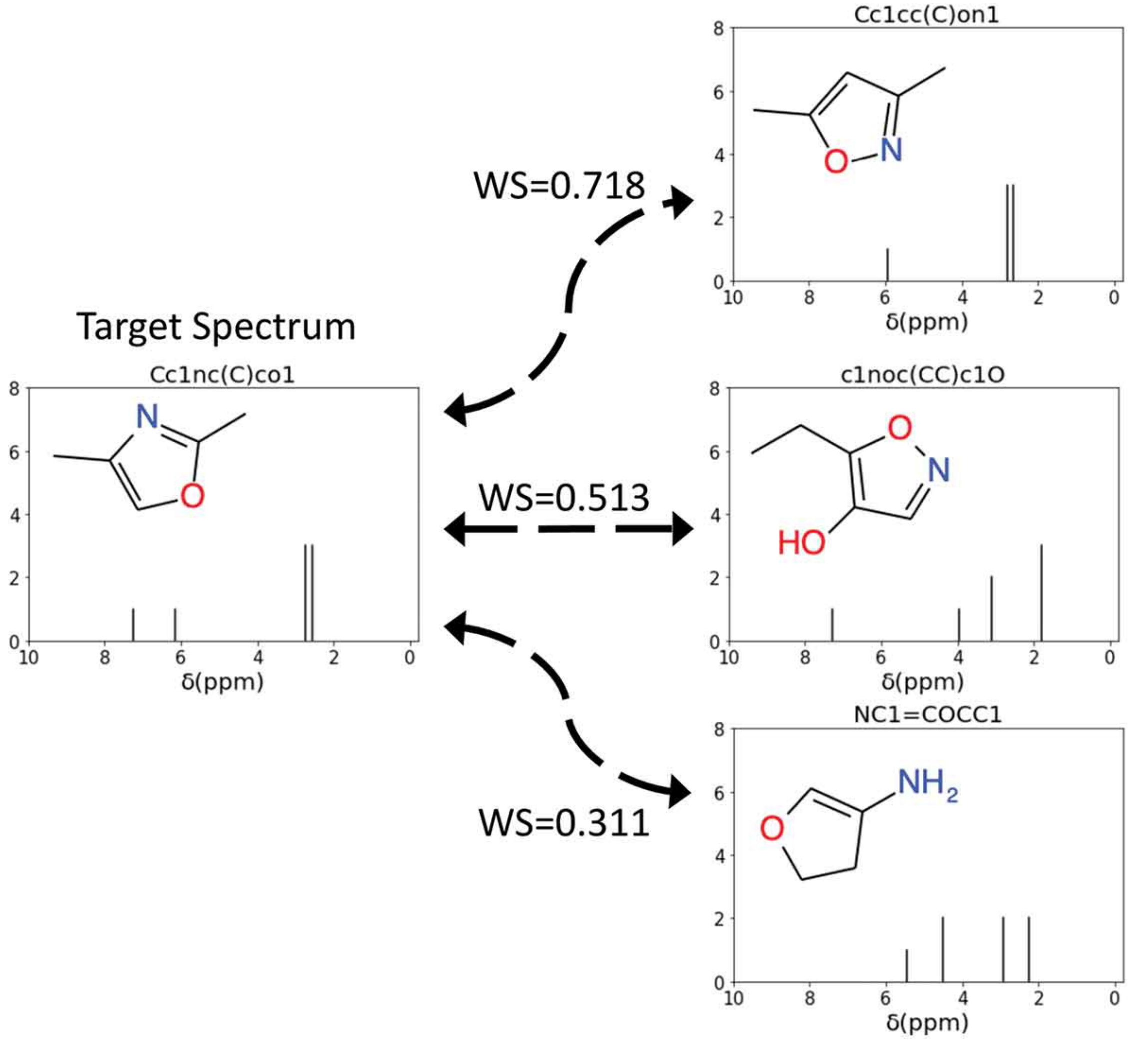


Figure 4.3: Wasserstein distance based evaluation function

Wasserstein Score is ranging from 0 to 1 where 1 represents two NMR spectra match perfectly and 0 means completely different. Closer a Wasserstein Score is to 1, more similar the two spectra and two molecules are to each other.

4.4 Trie enhancement of ChemTS

In this study, we remark several interesting characteristics about the NMR-TS: 1) Structure elucidation task from NMR spectrum has a large database as reference, 2) the computation of NMR spectrum for each generated molecule takes significant amount of time, 3) MCTS gains knowledge with the expansion of the tree, with more knowledge about the chemical space, MCTS will have a better chance to make good choice.

All these characteristic about NMR-TS lead us to an intuitive plan of optimization: do not start the MCTS from an empty tree, instead, preload the search tree with data knowledge.

In computer science, such prefix tree data structure is usually called trie [43].

For trie enhancement in NMR-TS, we execute the following steps:

- **Step 1**: compute all Wasserstein Score between target spectrum and each single NMR spectrum in the database. This function has a time complexity of $O(nS)$ where n is the number of spectrum-molecule pair in the NMR spectrum database, S is the time needed to computer Wasserstein Score between two NMR spectra. This process is needed for every unique target spectrum even the database is unchanged as the pair-wise score will be different for each target spectrum.

- **Step 2**: all pair-wise score for specific target spectrum will be sorted by decreasing order. Database molecules which have a top k Wasserstein Score will be selected for next step, k is the number of molecules we use for trie enhancement. We name k as trie size in the rest of the thesis.

- **Step 3**: we pre-build the Monte Carlo search tree with canonical SMILES representation of molecule structures selected in step 2 and insert them into the tree. Once a SMILES is inserted, we update the UCB score along the path with the Wasserstein score previously obtained.

A detailed algorithm of trie enhancement in ChemTS is described as pseudo-code in Appendix A.

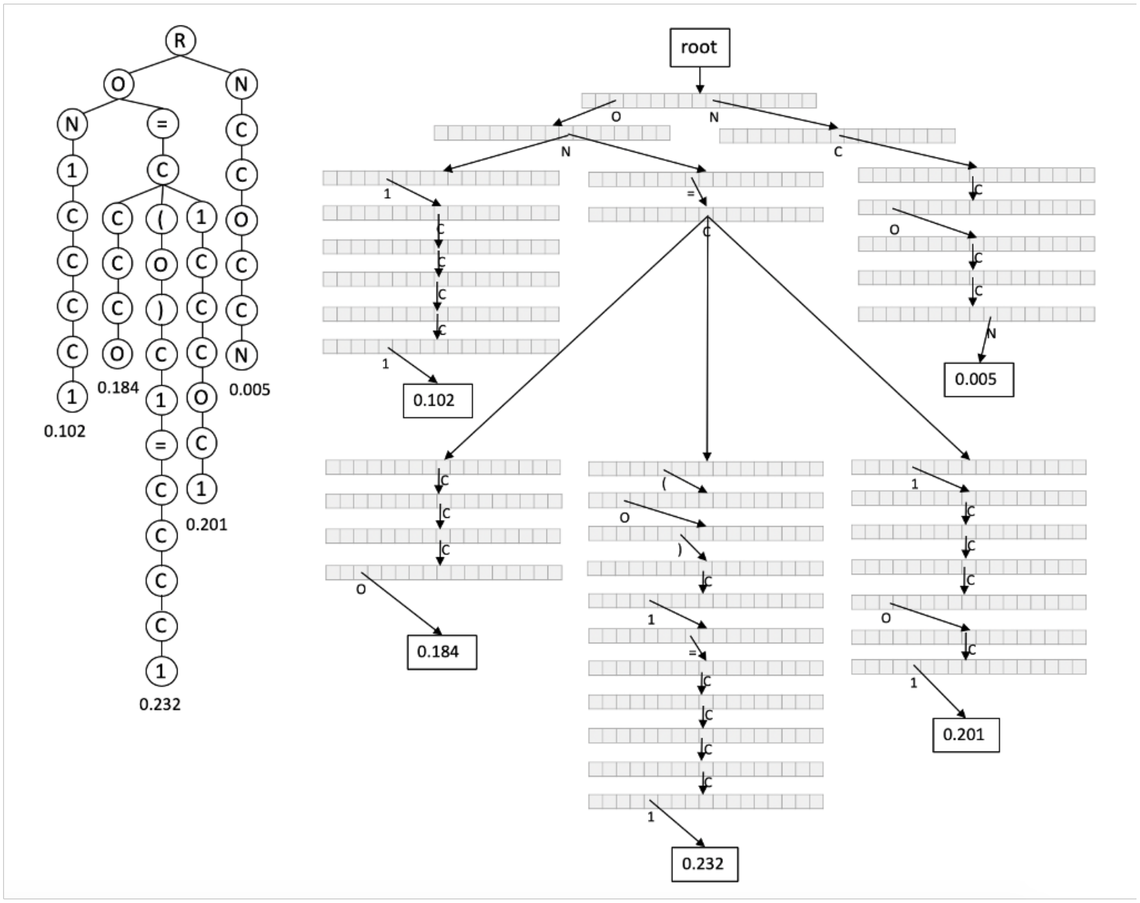


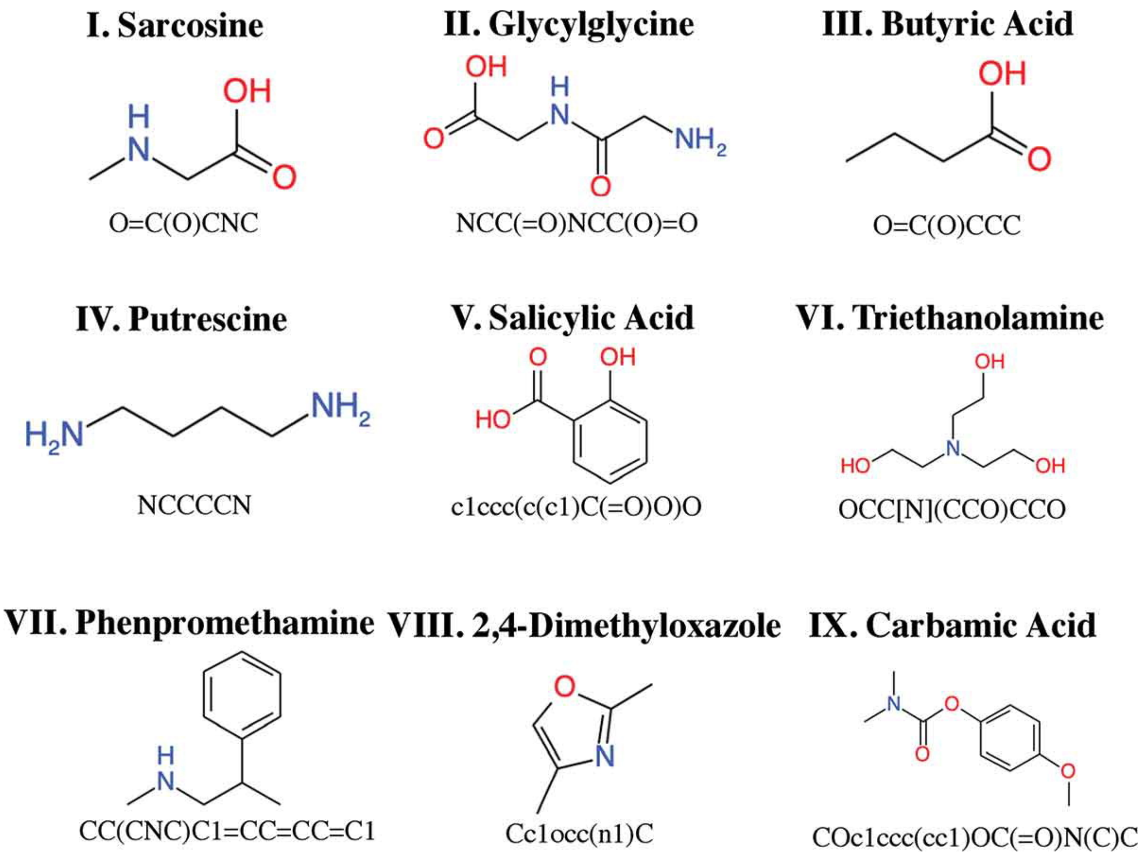
Figure 4.4: Illustration of Trie Enhancement of ChemTS

4.5 Database

To show the validity of our concept, we implemented NMR-TS and performed experiments to test the concept. For the database that we used in the experiment, we selected a SMILES database consisting of molecules with relatively small molecular weights. PubChemQC [44] is a free and open online database which contains over 3.5 million molecules. As PubChemQC also provides molecular properties computed by ab-initio calculation, we can therefore filter molecule structure based on properties such as the molecular weights. We selected molecules under 500 molecular weight, which is a suitable size for our study. We downloaded molecules in the form of SMILES strings with PCCDB-IDs from 1 to 138,895. We ran a selection on these 138,895 molecules to pick out the pure organic molecules that consisted of only C, H, N, and O. After selection, 10,548 molecules remained. Eight molecules were removed owing to the failure of the ^{1}H NMR spectrum computation. Charged molecules were also excluded. To verify that molecules not included in the database could be identified using NMR-TS, we removed the test molecules, which are described in the next section, from the database. Finally, 9866 molecules were used as the SMILES database. The database contained the following SMILES characters: O, c, 1, (,), C, =, N, #, n, 2, o, 3, and 4.

4.6 Testset

As a testset of molecules, we selected 9 molecules with similar size (under 500 molecular weight) to the PubChemQC database, all these 9 molecules are not included in our training set. Sometime database can have conformational isomers stored as different molecules, in this study we have explicitly checked the conformational isomers are not included. We performed NMR-TS calculation on the NMR spectrum of each individual molecules. A list of all 9 test set molecules are shown in figure 4.6.



I. Sarcosine

O=C(O)CNC

II. Glycylglycine

NCC(=O)NCC(O)=O

III. Butyric Acid

O=C(O)CCC

IV. Putrescine

NCCCCN

V. Salicylic Acid

c1ccc(c(c1)C(=O)O)O

VI. Triethanolamine

OCC[N](CCO)CCO

VII. Phenpromethamine

CC(CNC)C1=CC=CC=C1

VIII. 2,4-Dimethyloxazole

Cc1occ(n1)C

IX. Carbamic Acid

COc1ccc(cc1)OC(=O)N(C)C

Figure 4.5: Nine test molecules with their chemical structural formulas and SMILES representations.

4.7 Computational environment

The computations in this work were carried out on the RAIDEN supercomputer (RIKEN Center for AIP). For each test molecule, we execute the parallel version of NMR-TS on 20 CPUs for 100 hours. For each run, an average of 5451 molecules were generated by NMR-TS.

4.8 Baseline

To test the concept of NMR-TS, we set our baseline as the most matching molecule in the training set (database). To compute the baseline for every test molecule, we iteratively compute its Wasserstein Score against every molecule in the database and note the structure which obtained the highest score. We call the database molecule that obtained the highest score as baseline molecule and the highest score as baseline score. Baseline molecule and baseline score are different for different test molecule.

4.9 Results

We performed experiences of NMR-TS on all 9 test molecules. For 6 out of 9 test molecules, NMR-TS has found exactly the same molecular structure than the test molecule. For 3 molecules that NMR-TS did not find exactly the same molecule, in 2 out of 3 cases NMR-TS has found better molecule candidate comparing to baseline molecule. A detailed list of test molecules, baseline molecule and best NMR-TS candidate is shown in figure 4.6.

We have three major observation from this results. First of all, we can see several cases (**I. III. IV. VI.**) where NMR-TS successfully identified the molecule structure while baseline molecules is suggesting a relatively different structure. Secondly, For case **VII.**, NMR-TS have obtained equally good candidate as the database have suggested. After a review of candidate generated by NMR-TS for case **VII.**, we discovered that NMR-TS have generated a lot of different structure with a benzene ring and a C_4H_9NH tail. Although

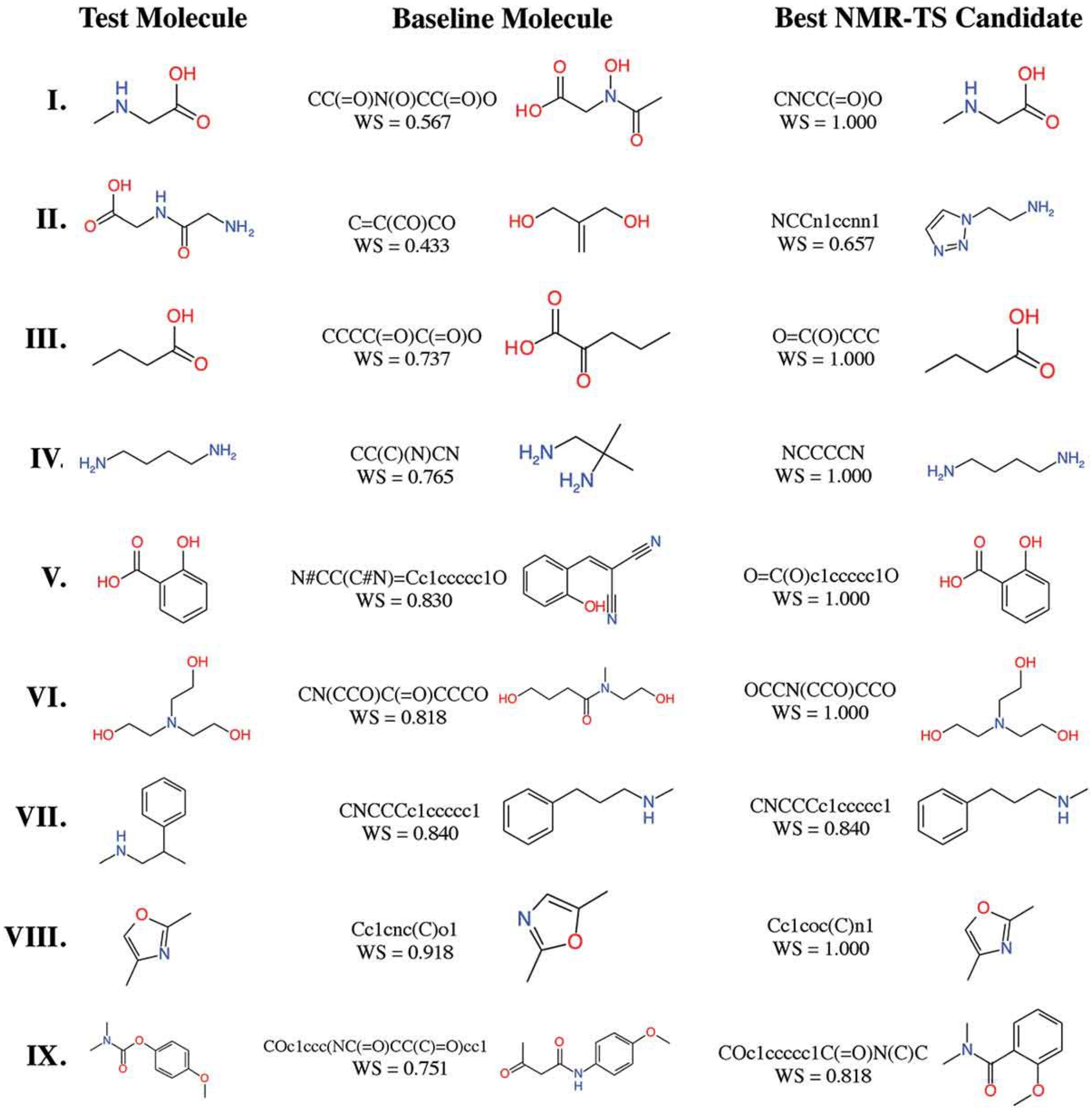


Figure 4.6: Test molecules, baseline molecules, and best candidate molecules generated by NMR-TS.

NMR-TS generated several different shape of C_4H_9NH tail, it could not find the exact match as the test molecule. We suspect that the correct tail substructure was not presented in the training set and therefore RNN was not able to predict similar structure. Lastly, we also noticed that for case **II.**, although NMR-TS obtained a better Wasserstein Score, the structure is actually more different from the target molecule which means NMR-TS entred a local minimum during search.

The evolution of Wasserstein Score in time under different size of trie enhancement for all 9 test molecules are shown in figure 4.7. The bold black dotted line represents the baseline Wasserstein Score for that molecule. Other lines represents the Wasserstein Score of the best candidate molecule obtained up to a certain hours of computation. Different colours of lines represent different size of trie enhancement. When trie size is 0 (blue line), it means no trie enhancement is used.

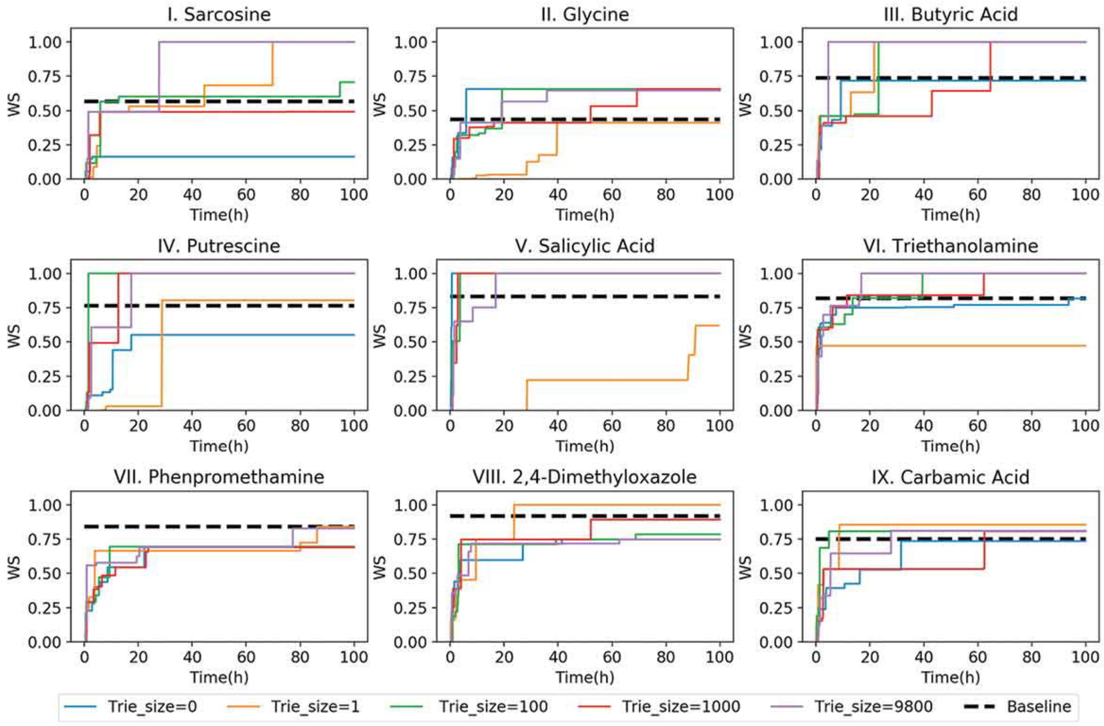


Figure 4.7: NMR-TS search results for target spectra of test molecules I–IX showing the best Wasserstein score (WS) as the function of time with different trie sizes.

What we can see from figure 4.7 is that for several test cases, multiple trie size run have succeed to identify the correct molecule structure. In general, a larger trie size lead to a faster finding of the correct molecule.

We created table 4.1 to summarize the target molecule found and the average of best Wasserstein Score obtained for different size of trie enhancement.

	Target Molecule Found	**Average of best WSs**
NMR-TS (Trie size = 0)	1 / 9	0.564
NMR-TS (Trie size = 1)	4 / 9	0.778
NMR-TS (Trie size = 100)	4 / 9	0.850
NMR-TS (Trie size = 1000)	4 / 9	0.837
NMR-TS (Trie size = 9800)	5 / 9	0.892
Database search (baseline)	0 / 9	0.740

Table 4.1: Correct answer rate and average Wasserstein score (WS) for each trie size.

On figure 4.8, we show average Wasserstein Score in time for all 9 test molecules. For all the trie sizes, the growth of the Wasserstein Score is mostly saturated within 40 hours. We can see that in general, largest trie size lead to highest average Wasserstein Score at the 100 hours time point. Larger the trie size is, higher average Wasserstein score reached. In addition, larger trie size will reach a certain average Wasserstein Score in shorter amount of time.

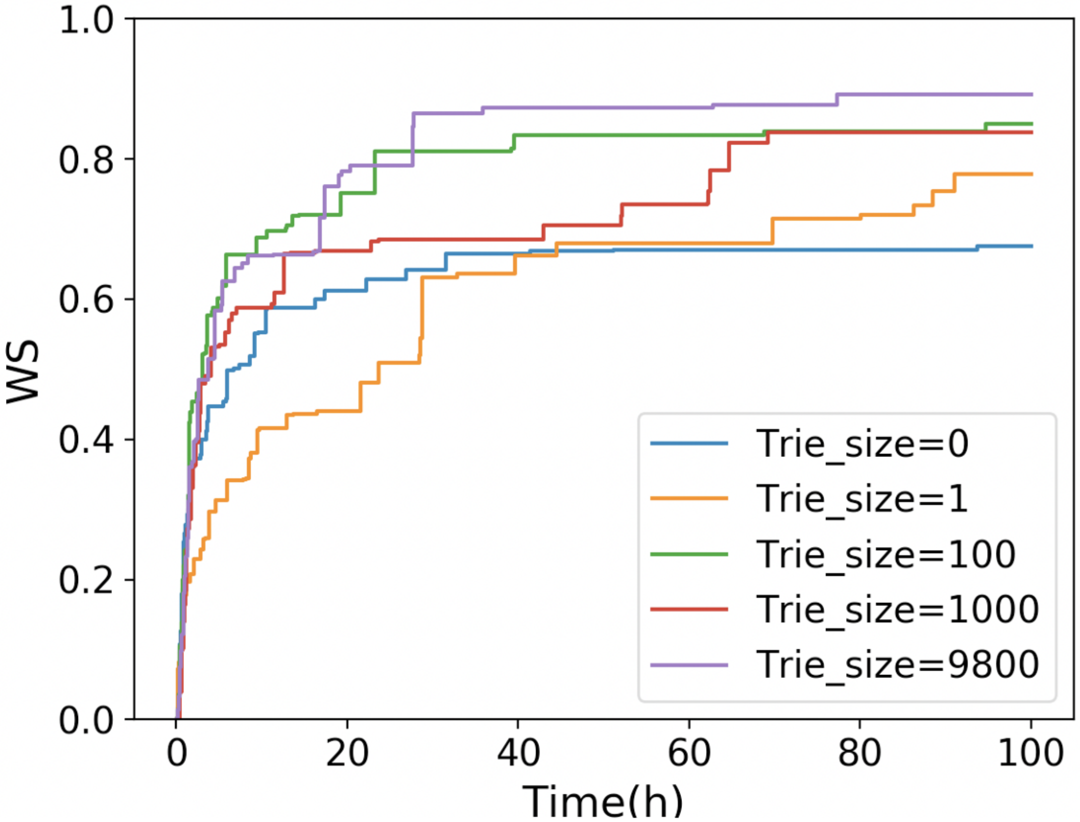


Figure 4.8: Evolution of the average Wasserstein score (WS) of the best candidates for the nine test molecules over time with different trie sizes.

On figure 4.9, we show the evolution of total number of the NMR-TS found candidates better than baseline for all 9 test molecules. For each trie size, the total number of candidates monotonically increased over time. NMR-TS generated more candidates with better scores than the baseline as the trie size increased. By comparing of the results for trie sizes of 0 and 1 in both figure 4.8 and 4.9, we see that a trie size of 0 was superior to a trie size of 1 from the viewpoint of generating more candidates with better scores than the baseline. In contrast, a trie size of 1 was superior to a trie size of 0 from the viewpoint of generating higher-scored candidates. A reasonable explanation for this phenomenon is that while the trie highlights the most promising branch in the search tree, the presence of the trie also restricts the exploration of other branches and thus reduces the overall diversity.

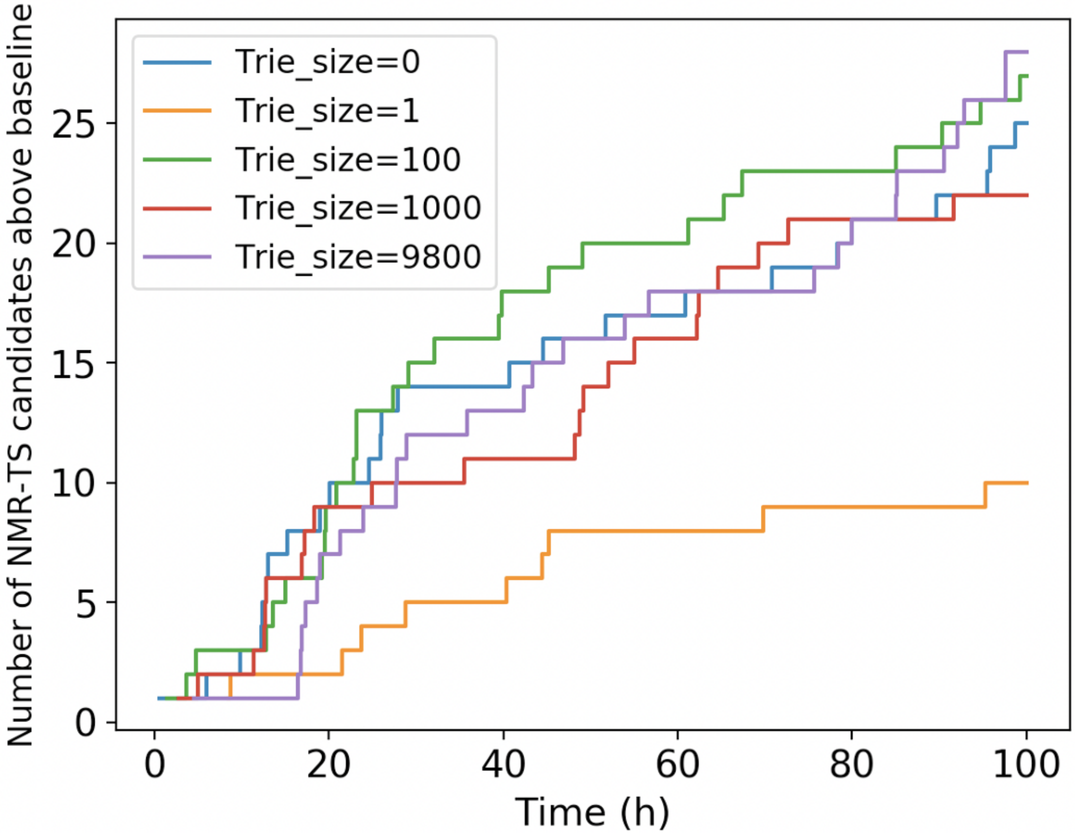


Figure 4.9: Evolution of the total number of candidates with scores better than the database baseline for all test molecules over time.

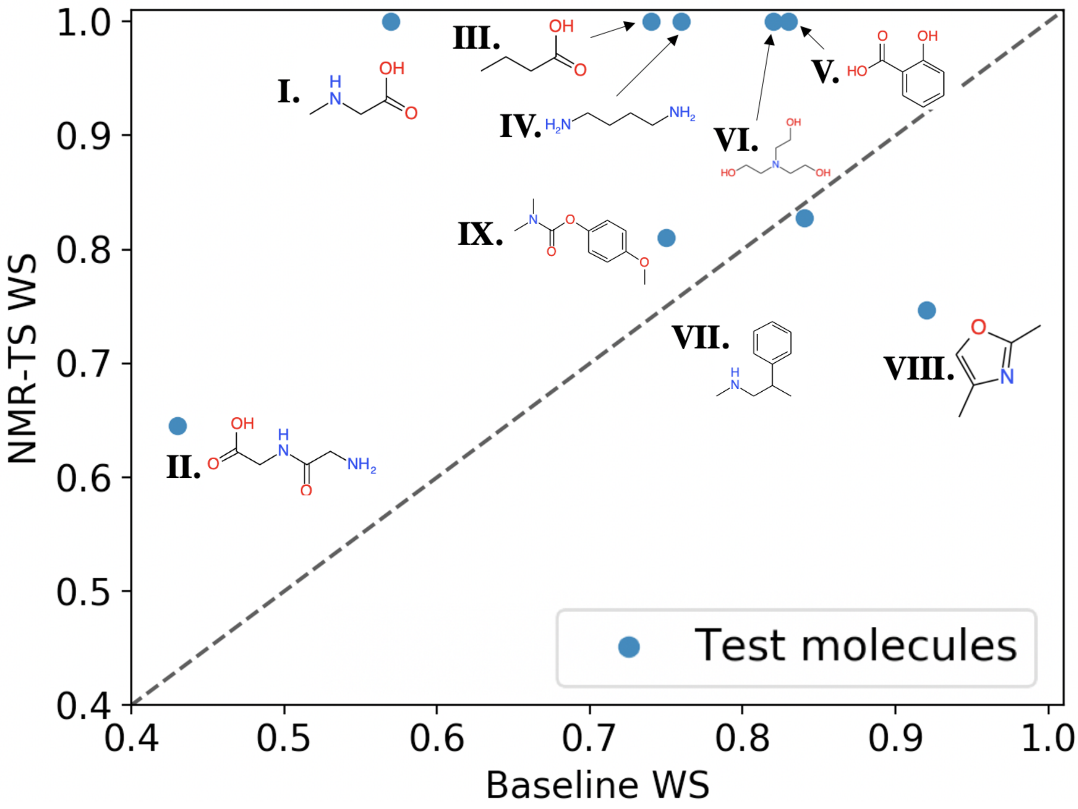


Figure 4.10: Comparison of the best candidate scores from the database search and NMR-TS. Trie size = 9800.

On figure 4.10, each point represents a test case. The horizontal axis is the baseline score and the vertical axis is the best NMR-TS candidate score under trie size 9800. If the NMR-TS obtains better result than the baseline, it should result a point above the diagonal dotted line on the figure. As we can see, on the vertical axis, several points reached score of 1.0 (**I. III. IV. V.** and **VI.**) indicates that NMR-TS succeeded in identifying the exact molecular structure. Although NMR-TS did not reach the baseline score for VII and VIII, these cases mainly fall on the extreme right side of the horizontal axis, which indicates that a good candidate already existed in the database. On the contrary, in cases where the baseline candidates poorly matched the target molecules (middle to left side of the horizontal axis), NMR-TS surpassed the baseline score.

Overall, we believe the prove-of-concept work returned cheering results in molecule identification task from NMR spectrum. In most of the cases, NMR-TS can either identify correct chemical structure out the NMR spectrum or suggest better candidate than database search. The failure cases are mainly related to either 1) local minimums created by the fact that different chemical structures can have similar NMR spectrum (we will further discuss this point in Chapter 5) or 2) the capability of generating various novel structures by *de novo* molecule generation model.

CHAPTER 5
METRIC BETWEEN NMR SPECTRA

In the previous chapters, we introduced about how NMR-TS can automatically identify molecular structure from a NMR spectrum. However, under the concept of NMR-TS, each NMR simulation suppose to generate a very accurate prediction about the actual experimental NMR spectrum of any molecular structure. This is hard to achieve with current NMR prediction methods. If no prediction method can predict an extensively accurate NMR spectrum for a molecule structure, the task of matching the experimental NMR spectrum to one of the predicted NMR spectrum becomes an information retrieval task. In this chapter, we will briefly review the current NMR spectrum prediction methods, metrics being used to compare NMR spectrum similarity and propose a potential solution to the metric issue of similarity measurement between NMR spectra.

5.1 NMR spectrum prediction

As one of the important physical properties of a molecule, an accurate prediction of NMR spectrum for chemical compounds is an essential building block for automated elucidation[45, 46, 47, 48, 49, 50, 51]. Traditionally, NMR spectrum can be computed using advanced first-principal calculations[50, 52, 53]. However, these methods have two major drawbacks: 1) they often requires human involvement to select hyper-parameters before computation, 2) the computational resources required are high which makes it almost impossible to apply on large scale systems. To overcome these limits, Bremser et al. [54] and Schaller et al. [55] has independently proposed empirical-data-based approaches based on reference molecule-spectra databases.

In the recent years, machine learning has shown great success in predicting outcome of a complicated system. The idea is to feed a machine learning algorithm with a representation of the chemical structure and its corresponding NMR spectrum. Earlier era of machine learning often represents each NMR active atom as a high-dimensional vector filled with hand-crafted features[20, 56, 57, 58, 59, 60]. Recently, with the rise of graph neural networks (GNN), studies started to represent the chemical structure as a graph where each atom represents a node and each edge represents a bond [61, 62, 63, 64]. As one study future, studies also merged message passing mechanism into GNN which results Message Passing Neural Network (MPNN) [23] solution or Graph Convolutional Neural Network (GCNN) [24]. Chemical shift prediction using a kernel machine based on DFT simulated data is also reported [25].

In most of the case, both first principal calculation and machine learning methods will produce a chemical shift value for every single NMR-active atom. However, the J-coupling effect is often not handled.

In figure 5.1, we show an example of Butyric Acid NMR spectrum, from experimental measurement, first principal calculation and GCNN.

As we can see, the original experimental NMR spectrum has rich details of J-coupling effect and slight noise. The ENSO simulated NMR spectrum calculated very accurately the chemical shift of each NMR-active atom. However, the J-coupling details are lost. In the GCNN predicted NMR spectrum, each peak is represented as a straight line as the results produce only chemical shift.

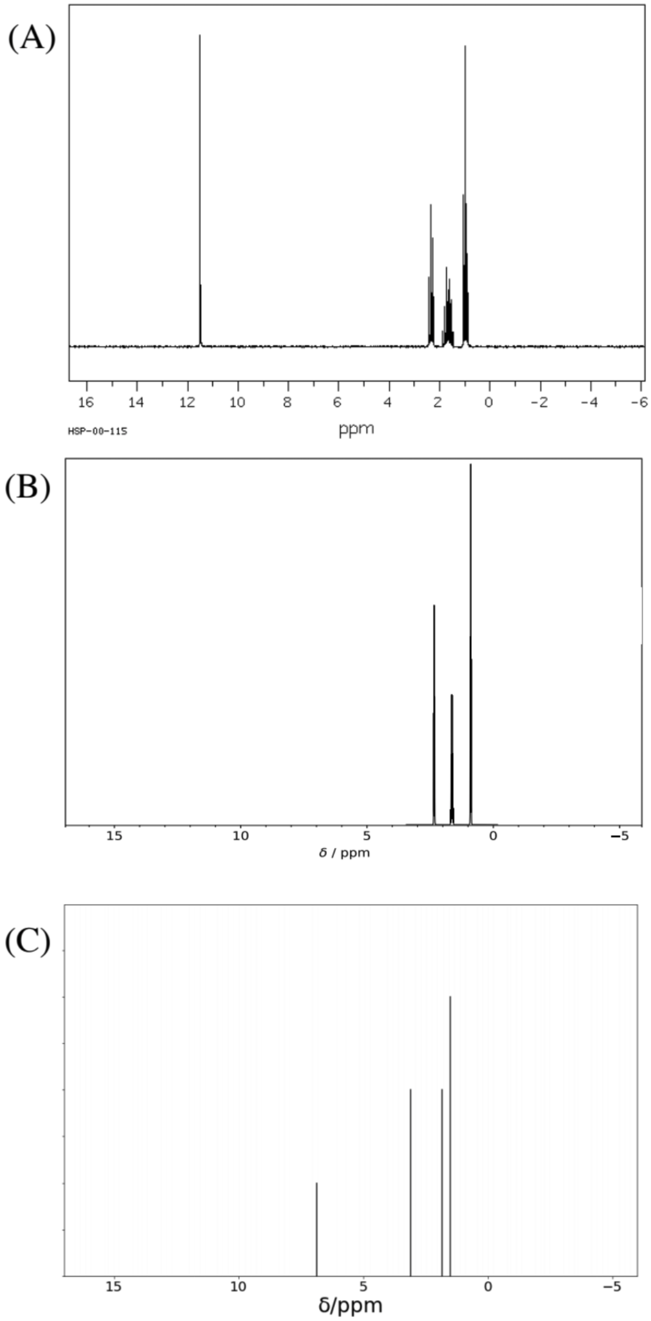


Figure 5.1: Comparison of (A) experimental ^{1}H NMR spectrum obtained from SDBSWeb database, (B) ENSO simulated ^{1}H NMR spectrum and (C) GCNN simulated ^{1}H NMR spectrum for Butyric Acid

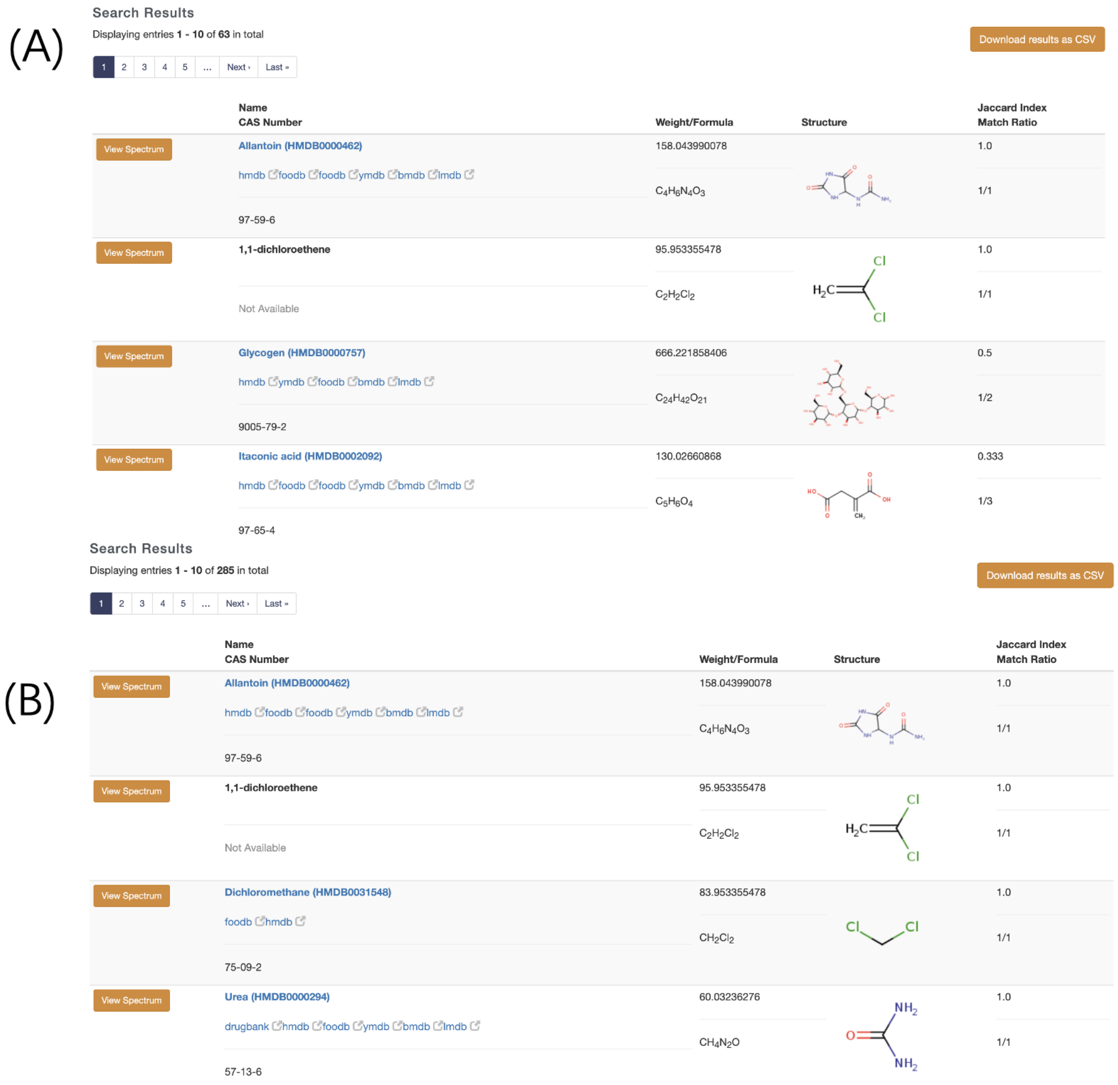
5.2 The tiny error between predicted and experimental NMR spectra

Although the accuracy of NMR prediction has increased significantly over the time. Most of recent studies still report an average error around 0.2 ppm for hydrogen chemical shift [24]. However, usually during database search, the tolerance of error between experimental spectra are set to 0.02 ppm by default. As one can easily imagine, if we enlarge the tolerance of error from 0.02 ppm to 0.2 ppm, the number of candidate structure that can potentially fit into the tolerance range will increase drastically.

In figure 5.2, we tried to search a random 1H NMR peak, 5.387 ppm for glucose, in The Humain Metabolom Database (HMDB). We tested both tolerance of 0.02 ppm, which is the default value of many similar database, and 0.226 ppm, which is the average error in ppm reported in recent chemical shift predictor using GCNN [24]. As a result, the database returned 63 candidate structures with 0.02 ppm tolerance and 265 candidate structures with 0.226 ppm tolerance. In addition, the top ranked candidates are different as well.

It is worth noting that, here, we are interested in finding the impact of tolerance on the number of candidates matched from database search. However, if multiple peaks are input to the search, most of the search engine will result candidates that match any peak amount input peaks. If the logic is set to find a structure matching all input peaks, which would be the case of structure elucidation task from monomer NMR spectrum, the number of candidates should be less than shown. However, the effect of tolerance on the number of candidates should be evident and should hold in both cases.

If multiple chemical compounds matches a NMR spectrum within the tolerance range, a question will raise naturally: which one is the ”correct” structure behind the spectrum?



Search Results

Displaying entries **1 - 10** of **63** in total

Download results as CSV

1 2 3 4 5 … Next › Last »

	Name CAS Number	Weight/Formula	Structure	Jaccard Index Match Ratio
View Spectrum	Allantoin (HMDB0000462) hmdb foodb foodb ymdb bmdb lmdb 97-59-6	158.043990078 $C_4H_6N_4O_3$		1.0 1/1
View Spectrum	1,1-dichloroethene Not Available	95.953355478 $C_2H_2Cl_2$		1.0 1/1
View Spectrum	Glycogen (HMDB0000757) hmdb ymdb foodb bmdb lmdb 9005-79-2	666.221858406 $C_{24}H_{42}O_{21}$		0.5 1/2
View Spectrum	Itaconic acid (HMDB0002092) hmdb foodb foodb ymdb bmdb lmdb 97-65-4	130.02660868 $C_5H_6O_4$		0.333 1/3

Search Results

Displaying entries **1 - 10** of **285** in total

Download results as CSV

1 2 3 4 5 … Next › Last »

	Name CAS Number	Weight/Formula	Structure	Jaccard Index Match Ratio
View Spectrum	Allantoin (HMDB0000462) hmdb foodb foodb ymdb bmdb lmdb 97-59-6	158.043990078 $C_4H_6N_4O_3$		1.0 1/1
View Spectrum	1,1-dichloroethene Not Available	95.953355478 $C_2H_2Cl_2$		1.0 1/1
View Spectrum	Dichloromethane (HMDB0031548) foodb hmdb 75-09-2	83.953355478 CH_2Cl_2		1.0 1/1
View Spectrum	Urea (HMDB0000294) drugbank hmdb foodb ymdb bmdb lmdb 57-13-6	60.03236276 CH_4N_2O		1.0 1/1

Figure 5.2: 1H NMR search result of peak 5.387 from HMDB. In (A), the tolerance of 1H NMR is 0.02 ppm. In (B), the tolerance of 1H NMR is 0.226 ppm

5.3 Similar NMR spectra could come from different molecules

In the current NMR-TS concept, the Monte Carlo search tree optimizes the Wasserstein distance based score. The underlying assumption of this search is that, more similar two NMR spectra visually are, more similar the chemical compounds behind them are. However, this is not always true. It could happen that significantly different chemical structures could accidentally produce similar spectrum as different functional groups overlap on some area of the NMR spectrum.

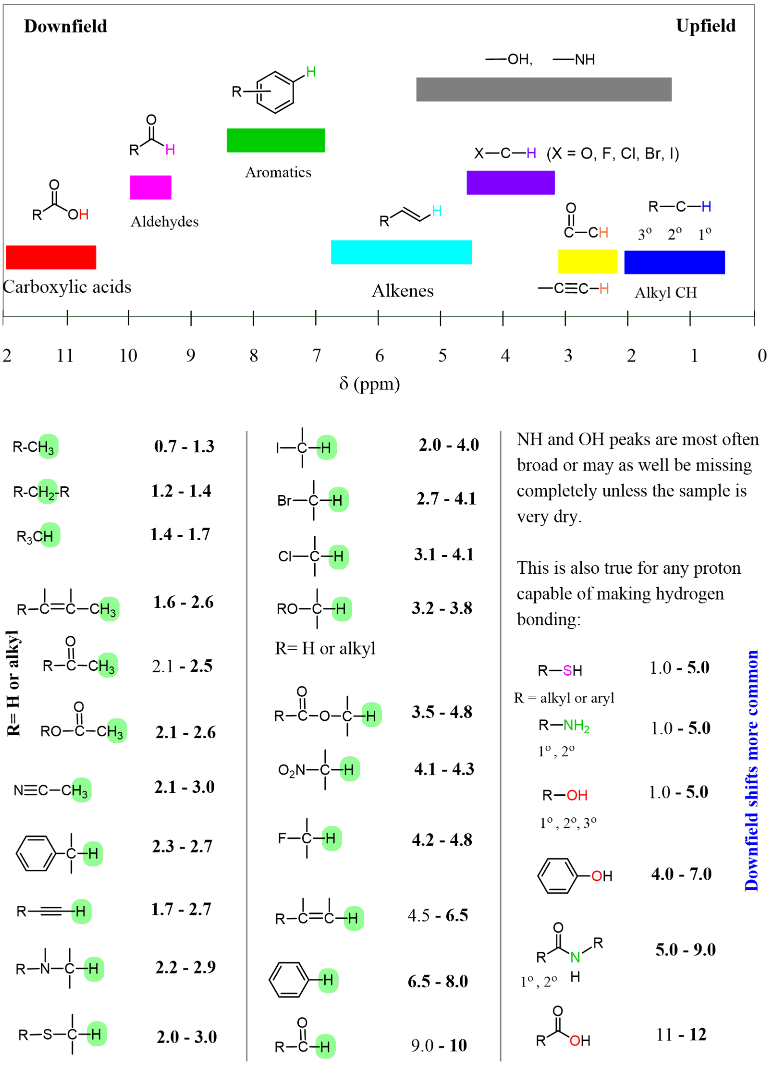


Figure 5.3: 1H NMR Shift Table

To measure the distance between two NMR spectra, several studies has been conducted previously. Studies such as cross-correlation[65, 66] or spectra-intersection [67] were the firsts applying a pairwise comparison between spectrum vectors. These concepts require significant amount of computational resources and therefore not suitable to apply on large quantity of data such as the scenario of a database. Another approach using a gradual division of binning is also proposed and demonstrated good classification accuracy[68].

To measure the distance between two NMR spectra, several studies has been conducted previously. Studies such as cross-correlation[65, 66] or spectra-intersection [67] were the firsts applying a pairwise comparison between spectrum vectors. These concepts require significant amount of computational resources and therefore not suitable to apply on large quantity of data such as the scenario of a database. Another approach using a gradual division of binning is also proposed and demonstrated good classification accuracy[68].

The integral of chemical shift peaks are binned into a successively smaller area and the integral value of each bin are compared. Theoretically, two exactly same NMR spectra should have same integral value on each single bin. This method doesn't rely on peak tables and is computationally more efficient. In [69], a tree-based similarity measurement is also proposed. This method builds a tree based on the proximity between peaks and then compare the similarity between two trees built from two input NMR spectra. It does not rely on peak-picking or any pre-treatment of the data.

5.4 Metric learning on NMR spectrum similarity measurement

For most of the distance metrics used for NMR spectra comparison are rule or formula based. Usually, the chemical structure implication of the chemical shift and the implication of different sources (NMR instruments or NMR predictors). As a result, the information retrieval process of matching an NMR spectrum to a spectrum stored in the database is not always successful, especially when the query spectrum comes from a different source than the database stored one. A possible way out for both of the above addressed issues is to create a customized metric that measures the similarity with consideration of the chemical structure implication of each peak and/or the source of the NMR spectrum.

Metric learning is a sub-area of supervised machine leaning, it is closely related to classification and regression, but the goal is to measure a similarity distance between two inputs. Notably, it aims to learn distance metric from the data provided [70]. With adequate amount of data feed into the algorithm, we can obtain a unique model that can predict similarity distance based on the data learned.

Among various metric learning concept proposed, one of them drew our attention, which is the Siamese Neural Network (also called Twins Neural Network). Siamese Neural Network is a type of artificial neural network that uses the same weight for different classes of input and convert input vectors into more comparable output vectors. This type of neural network can achieve classification tasks such as information retrieval. In many traditional classification methods, we use various model to compute a class label prediction from input vector. However, in Siamese Neural Network, we project different input vectors into an output space and try to put data points from the same class as close as possible in the output space while maximizing the inter-class distance. This later part of task is often achieved by introducing Triplet Loss as training loss function.

$$L = max(d(a,p) - d(a,n) + margin, 0) \tag{5.1}$$

In the triplet loss function, the loss L is depending on the distance between anchor data and the positive data $d(a,p)$, the distance between anchor data and the negative data $d(a,n)$ and a $margin$ which is a hyper-parameter defines the minimum amount of dissimilarity there should be for different classes.

Many application using such Siamese-Triplet Loss combination has been proven successful [71] in few-shot learning tasks such as person re-identification [72]. In this section, we try to apply similar techniques on NMR spectrum classification.

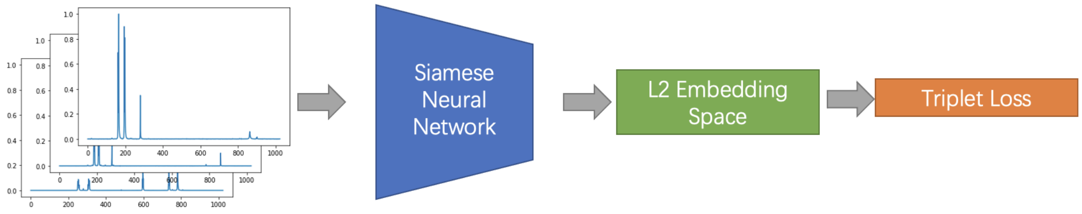


Figure 5.4: Concept of applying metric learning on NMR spectra classification

As shown on Fig 5.4, under this concept, we consider the task of matching one NMR spectrum to the closest one in a NMR spectra database as a re-identification task (or information retrieval task). Each molecule structure is a single class, different sources of NMR spectrum (predictors or experiments) are like different photos of the same molecule. Our task, is to train a neural network model that can projects input vectors into an output space, where it minimize the intra-class distance and maximize the inter-class distance.

5.5 Input Representation Form

In our study, we tried two different representation from of the input NMR spectrum. The first input representation we used is called the ”intensity representation” where we feed the original NMR spectrum as a one-dimensional intensity image. The second input representation is called the ”bin representation”, where we are inspired by the previously proposed bin-representation distance metric. In this section, we will introduce in detail about both representation form.

5.5.1 Intensity representation

For intensity representation, we take NMR spectrum as a one-dimensional intensity image (where we only have one intensity instead of three different colors). As the original NMR spectrum is a continuous curve and our neural network can only take discrete inputs, we need to separate the continuous curve into a fixed amount of bins and take the average intensity value of each bin. This number of bins is similar to the concept of resolution for classical image processing. In our study, we defined the resolution to 1024 for the convenience of matching a previous study[69].

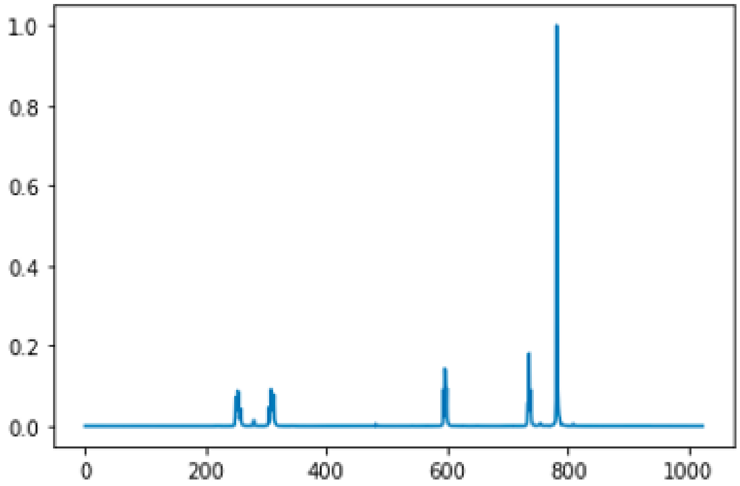


Figure 5.5: Example of intensity representation form of NMR spectrum, the horizontal axis represents the index of input intensity

5.5.2 Bin representation

The bin representation is inspired from a previous NMR spectrum similarity metric [68]. In this study, L.Bodis et al. used a successive binning way to measure the similarity between two NMR spectra. At each iteration, both NMR spectra will be split into a number of bins, this number of bins increase by one for each new iteration. After binning, the integral of each bin is calculated and compared with the bin at the same position on the other NMR spectrum, the minimum value of these two bins are conserved and used to compute the final similarity distance. In the final calculation of the similarity distance, the algorithm will take into consideration of every bins generated in all iteration. However, later iteration will have a higher impact on the overall score as their bins' range are smaller and therefore represents a closer match.

More formally, we denote V as the input vector to the neural network

$$V = B_1 \oplus ... \oplus B_n \tag{5.2}$$

where

$$B_k = I_{k1} \oplus ... \oplus I_{kk}$$

with I_{ki} represents the integral of ith bin under the splitting of k bins. n is a hyper-parameter that we have set to 50 to make the total number of bins to 1250 which is similar to Intensity Representation.

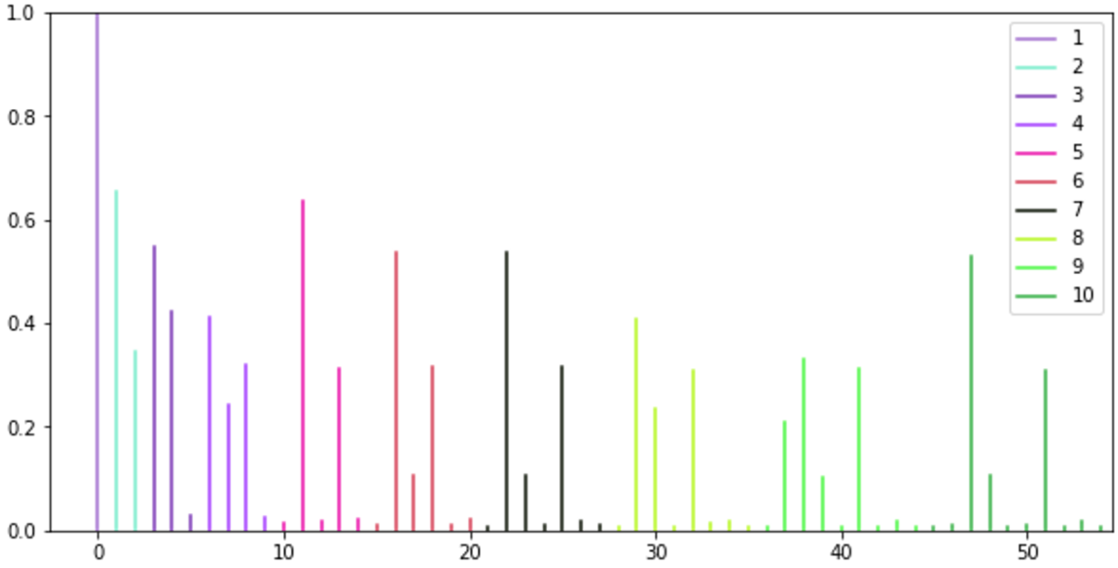


Figure 5.6: An illustration of bin representation, only include first 55 bins starting from the left

We illustrate the concept in figure 5.6. On this figure, each different color represent a different series of B_i. On the left side, the first B_i is B_1 where the integral of the whole NMR spectrum is calculated. Next to the right, the B_2 split the NMR spectra into two area and the first area contains about 63% of the total integral and the second area contain about 37% of the total integral. The sum of each B_i are all equal to 1.

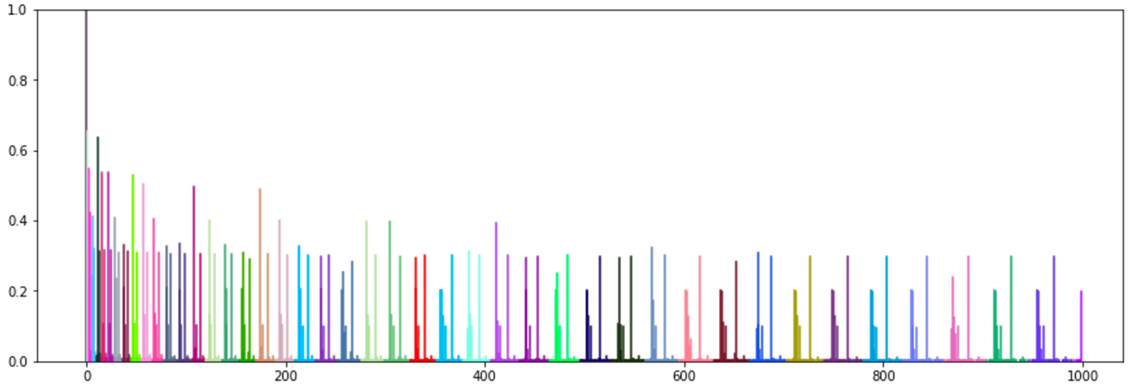


Figure 5.7: An example of full bin representation, all B_i are marked in a different color

5.6 Triplet mining strategy and margin

Under the concept of triplet loss, we need to pull the distance between anchor data point and the positive data point as close as possible and push the distance between anchor data point as far as possible from negative data point. To make this work, we have to first select three element to build a triplet: 1) an anchor data, 2) a positive data and 3) a negative data. A critical process during training is how to select this training triplet. This selection process is called triplet mining. In this section, we will discuss about some common triplet mining strategy and explain our approach.

5.6.1 Easy, hard and semi-hard triplets

For any possible triplet that we from a training dataset, depends on the relative distances between its three components and the margin we set as hyper-parameters, a triplet can be categorized into three different categories:

- **Easy triplet**: triplet with a loss of 0, this usually means that $d(a, n) - d(a, p) > margin$

- **Semi-hard triplet**: triplet where $d(a,p) < d(a,n) < d(a,p) + n$, this usually means the negative point is not closer to the anchor, but is not far enough depending on the margin.

- **Hard triplet**: triplet where $d(a,n) < d(a,p)$, meaning that negative point is closer to anchor than the positive point

In our case, most of the NMR spectra are significantly different with other NMR spectra if they are not coming from the same class (same chemical structure). If we randomly select a triplet and use all triplets generated, most of these triplets will be easy triplets which won't create any loss. This will become a significant waste of computational resources during training as null gradient is not useful for learning. Therefore, what we choose to do is to use only semi-hard and hard triplets.

5.6.2 Online and Offline mining

The distance between anchor, positive and negative points is dynamically changing with the update of weights in the neural network. In the last section, we mentioned that we only use semi-hard or hard triplets for training. But with the training process going on, the definition of one triplet being easy, semi-hard or hard is constantly updating as well. This property force us to recreate triplets after an epoch and therefore an efficient triplet creating strategy is desired.

- **Offline mining**: In offline mining strategy, we compute all the distance for all possible triplets before each epoch. Concretely, we produce an enumeration of all combination of (i_a, i_p, i_n) and compute all the distance pairs for each combination. This strategy is extremely computationally heavy especially if we have a large data set for training. In addition, most of the triplets generated are not valid because the anchor data has to belong to the same class as the positive data and the negative data should belong to a different class.

- **Online mining**: The online mining strategy on the other hand, don't require to compute all possible triplet combination. Instead, we select every component of the triplet "on the fly": 1) first we select a random class as anchor class and select a random input from this class as anchor 2) select a different input from the anchor class as positive, 3) select a random input from a different class as negative. This mining strategy will only generate valid triplets and is therefore more efficient.

In our experiments, we selected online mining strategy to save computational resources.

5.6.3 Margin

In all of our experiments, we set our margin to 0.2 which is an empirically good fit from several attempts.

5.7 Dataset

To test and compare our concept of apply siamese neural network + triplet loss for NMR spectrum classification, we need a NMR spectrum dataset that has multiple molecule structures and has multiple sources generated NMR spectra for the same molecule structure. Particularly related to triplet loss, we need at least three NMR spectra per molecule structure: one as anchor data, one as positive data and one as test data. In [73], researchers shared a dataset that contains 1000 different molecule structures and there are five different sources of NMR spectrum for each molecule structure. The five different sources are: experimental NMR spectrum from Maybridge catalogue and NMR spectra of the same molecule structure from four different NMR spectrum predictors [20, 22, 57, 74]. As the authors keep the data anonymous to a particular predictor, we name all five sources of data respectively as experimental, predictor1, predictor2, predictor3 and predictor4.

In the original dataset, all intensity values are given in a resolution of 1024 and all intensities are ranging from 0 to 10,000,000. We have normalized all intensity value between 0 and 1 to fit as neural network input.

We choose to use experimental spectrum and predictor 2 to 4 as training data, and we use predictor1 data as test data from classification tests.

5.8 Neural network model and training

We implemented siamese neural network for different input representation using Keras and Tensorflow. The network mainly consists of two part: 1) a multi-layer Convolutional Neural Network (CNN) part that extract graphical features from the input vector, 2) a fully connected neural network which projects input vector onto a customized output L2 restricted space.

A detailed description of siamese neural network structure that we used in this study is described in Appendix B.

5.9 Baseline and metrics

To make our study comparable to [73], we used Mean Reciprocal Rank (MRR) as the metric that evaluation the goodness of each similarity metric.

The MRR is defined as:

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank_i} \tag{5.3}$$

where n is the index of queries and $rank_i$ is the rank of the correct matching index for query i.

MRR is a metric that favorites a match on very high rank (e.g. rank 1 or rank 2). A relatively low rank (e.g. rank 100 or rank 400) has numerically little impact on MRR. Concretely, the MRR don't care much about ”how wrong a classification goes”, whether it is rank 50 or rank 400 makes little difference. That is why we also introduce Average Rank (AR) as another metric to evaluate the classification performance:

$$AR = \frac{\sum_{i=1}^{n} rank_i}{n} \tag{5.4}$$

where n is the index of queries and $rank_i$ is the rank of the correct matching index for query i.

For baseline methods, we were able to re-implement the bin similarity method and obtain adequately good performance. So we choose this method as our baseline method to compare with.

It worth noting that, we have also tested tree similarity method [69] as a potential baseline method. Particularly in the tree similarity metric, a set of hyper-parameters need to be defined. Although we tried the default setting of hyper-parameters and made several different attempts, we were not able to reproduce a comparably good MRR performance as claimed in the original paper. That's why we did not include this method as baseline reference.

5.10 Results

We have trained our neural network on experimental, predictor2, predictor3 and predictor4 data for 1000 different molecules. To evaluate, we inference the NMR spectrum from predictor1 to the embedding space and measure its distance against all training points in the embedding space. We rank all classes for every testing data point by the proximity order. The $rank_i$ of data i is the rank of the correct class appeared in this proximity order. With ranks of all 1000 test data, we can then compute MRR and AR for all similarity distance model.

In figure 5.8 to figure 5.11, we show examples of hardest triplet before training and after 12 hours of training for both Intensity Representation and Bin Representation.

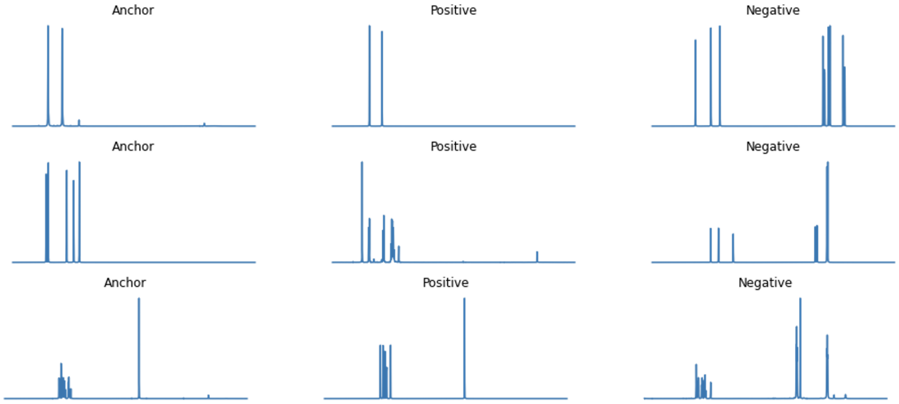


Figure 5.8: Examples of hardest triplets before training for intensity representation

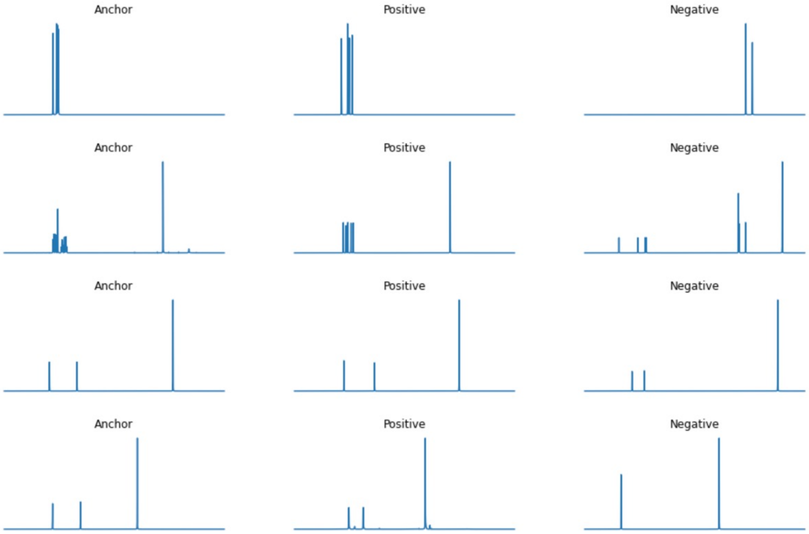


Figure 5.9: Examples of hardest triplets for intensity representation after 12 hours of training

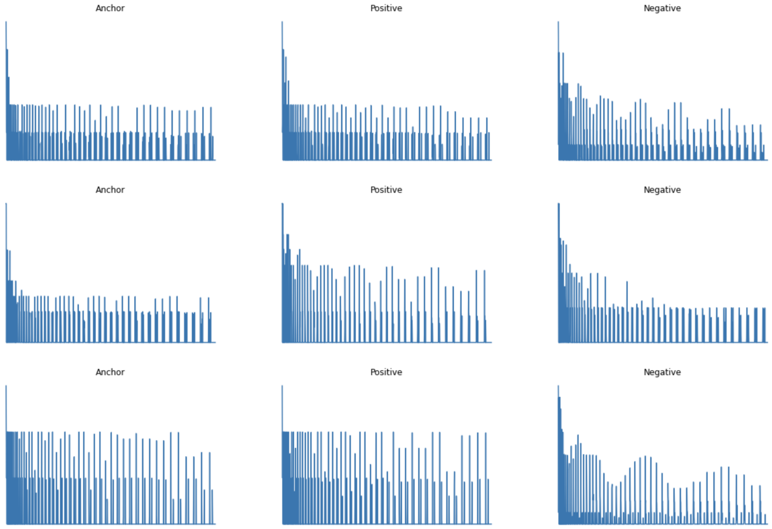


Figure 5.10: Examples of hardest triplets before training for bin representation

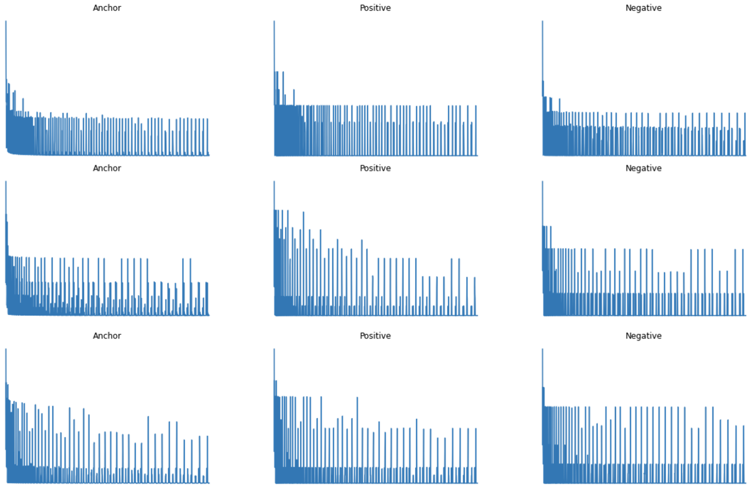


Figure 5.11: Examples of hardest triplets for bin representation after 12 hours of training

As we notice from the above figures, for both Intensity Representation and Bin Representation, the hardest triplets are comparably more ”difficult” after 12 hours of training. This proves that the gradient decent process converged and the training has significantly helped the neural network put different classes in different positions of the L2 embedding space. Only very confusion cases (i.e. visually close cases) can produce a semi-hard or hard triplet. In fact, for most of the hard triplets, it is even difficult for human beings to discriminate between positive and negative data because either the negative spectrum is too similar to the anchor spectrum or the positive spectrum is too dissimilar to the anchor spectrum.

We show the result of MRR and AR in Fig 5.10. In terms of input representation we tested, the Intensity Representation and the Bin Representation are formally defined in the previous section. What we have noticed is that the data points that they succeed to classify are not overlapping a lot. In another word, the data points that Intensity Representation are doing great are not necessarily correctly classified for Bin Representation even that Bin Presentation has higher score depends on our metrics. This drives us to test a mixed version of Intensity Representation and Bin Representation, namely Intensity + Bin Presentation, which is a concatenation of both individual representation vectors. We compare all three input representation results against our baseline reference method: bin similarity method.

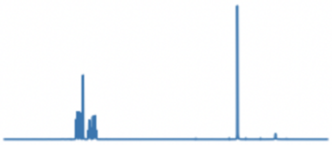
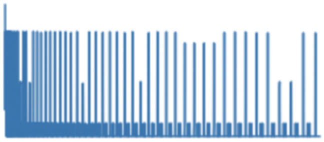
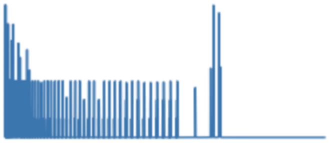
	Input Vector Visualization	Input Shape	MRR	AR
Intensity Representat ion		(1024, 1)	0.39	14.12
Bin Representat ion		(1250, 1)	0.59	**5.22**
Intensity + Bin Representat ion		(2274, 1)	0.55	**5.34**
Bin Similarity	N/A	N/A	**0.64**	9.97

Figure 5.12: Comparison of MRR and AR for different metric measurement methods. For MRR, higher score is better. For AR, lower score is better.

In terms of MRR, the baseline method obtained the best performance. Among different neural networks, the Bin Representation obtained the best performance which is close to the baseline.

In terms of AR, both Bin Representations and Intensity + Bin Representations obtained an average AR around 5.3 which is significantly better than the baseline. It suggests that neural network has a better robustness to classify difficult data points.

Overall, we believe that the successive bin splitting pre-processing is a significant boost to the classification task. With the rest of the neural network remains similar, changing the input vector from Intensity Representation to Bin Representation has significantly improved the performance on both MRR and AR.

5.11 Discussion

In this study, we 1) addressed the necessity of a data-driven distance metric between NMR spectra of different sources, 2) proposed a concept using metric learning which could be a potential solution to the issue and 3) implemented a prove-of-concept of metric learning method onto NMR spectrum classification tasks.

Although classification between NMR spectrum having the same molecule structure but different sources is an important task, other metric related issue should also be considered to solve using metric learning type of approach. An example of metric issue that we could not explore during this thesis is that: can we predict a molecule similarity distance based on inputs of their NMR spectra?

With our prove-of-concept study, we were able to create a customized metric that can correctly classify known molecule's NMR spectrum from unknown source. The performance prove-of-concept implementation is at least comparably good as the best baseline that we can find. However, we believe that there are still rooms for the performance to growth:

- **Training data**: In facenet, the smallest training set used contains about 2.6 million image. Increasing training set from 2.6 million image to 260 million image improves the overall classification accuracy from 76.3% to 86.2%. In our training set, there are only 1000 molecules used which represents 1000 class, with 4 ”images” per class which is relatively small for training. Using a larger set of training data might increase the performance.

- **Multi-modal data**: To increase the training set of data, the major obstacle is that the data of experimental NMR spectrum - molecule structure data has a limited quantity. Although most of the NMR predictor works in a time-efficient way, several sources of NMR spectrum might not be easily accessible such as DFT computed NMR spectrum. This inconsistency of input data for individual molecule structure enters the category of multi-modal learning. Related method could be considered in the future.

- **Transformer encoder**: Recently, the multi-head attention mechanism became a hot topic in Neutral Language Processing (NLP) applications. One of the advantages of Transformer is its ability to learn the long range interaction between words. For NMR spectrum, if we see every single NMR spectrum as a set of peaks, the concept of spectrum and peaks are similar to the concept of sentences and words. So instead of using CNN layers to extract features from NMR spectrum, we might try to use Transformer encoder to do the task and learn more about interactions between peaks.

CHAPTER 6
CONCLUSION AND OUTLOOK

6.1 Summary

In this thesis, we designed NMR-TS as a system that can automatically elucidate chemical structure from NMR spectrum. The system use a combination of *de novo* molecule generator and quantum chemical calculations. Depends on our testset of molecules, under the context of identifying a chemical structure unknown to the database, NMR-TS could identify better candidate comparing to the database in most of the cases and at least as good candidates as database in all cases. NMR-TS successfully identified 6 out of 9 molecules without any prior knowledge to neither the NMR spectrum measured nor to the chemical structure.

As a molecule generator, ChemTS is proven to be a useful and popular generator for *de novo* molecule design. In this thesis, we suggested a enhancement for ChemTS that can integrate of prior database knowledge into the search. In the case of NMR-TS, we demonstrated that such enhancement can 1) reduce the search iteration towards a certain level of score, 2) increase the peak performance of the Monte Carlo tree search, which in the context of NMR-TS, is the success rate of identify the molecular structure from NMR spectrum. The trie enhancement can be applied not only on NMR-TS, but also on other tasks using ChemTS and a time consuming scoring system.

NMR-TS automatically explore the metric space of a target spectrum. The metric we use is built on Wasserstein distance based score. NMR-TS does not use any domain specific knowledge related to 1H NMR spectrum, which suggests it would be possible to apply similar method to other spectroscopy techniques such mass spectroscopy, 13C NMR, IR and UV-vis spectra with adequate metric designed.

A higher level aim for NMR-TS is to set a prove-of-concept study for using *de novo* molecule generator for tasks other than a numeric property goal. This study has showcased an example of optimization towards a qualitative target.

In addition, during the course of NMR-TS, we found that most of the *de novo* molecule generator are aiming to find ”good” molecules. As the chemical space is extramly large. We theoretically has a lot of ”good” molecule candidates. However, in the application of molecule identification, we have only one (in case of SMILES, a handful as different SMILES can represent same chemical structure) ”correct” answer to find. This creates new challenges for molecule generation studies: 1) finding a ”correct” answer in the tremendous chemical space requires an much higher level of smoothness for the model projected chemical space, 2) finding one single answer out of a chemical space requires the optimization algorithm to have good balance between global and local optimization. We believe NMR-TS demonstrated the necessity of a *de novo* molecule generator that can find one single specific molecule structure.

We have also explained the importance of an accurate metric between NMR spectra generated by different sources. We proposed a potential solution for this issue using siamese neural network and triplet loss. Our prove-of-concept work showed comparably good results as our baseline method with a small training set. We believe there are room of improvement in terms of performance with larger training set and/or more sophisticated neural network structure.

6.2 Future work

There are several limitations and room for improvements regarding the current version of NMR-TS. First of all, SMILES as a representation form of chemical structure can not represent several chemical features in 3D such as axial chirality. In our study, the potential chirality space and conformer space of the chemical structure is not discussed. To tackle these spatial feature of chemical compounds, the *de novo* generator must be able to design these features. Recently, *de novo* molecule generators supporting 3D structures are started to emerge[75].

Another limitation about NMR-TS is that it is set to the context of mono-molecule situation. Under multiple molecule presents in the sample of NMR spectrum, which often is the case in reality, chemical shifts of several different molecules will overlap on the same spectrum. It is important to find out the corresponding relationship about which chemical shift belongs to which chemical structure. There is study about peak separation for this purpose that could be a potential boost[76].

There are various potential that the concept of NMR-TS could make use of. For example, as an automated elucidation method for *any* molecule, it could be used as a qualitative measurement in automated robotic synthesis system. Future work should focus on mainly three directions: 1) extracting and identifying individual molecules from a multi-molecule spectrum, 2) creating a customized metric or better prediction method for NMR spectrum to close the gap between experimental spectrum and simulated spectrum, 3) integrate spectrum specific knowledge into the generation process to reduce the search space and improve time efficiency.

For distance metric between NMR spectra, future work could develop into several directions. In our prove-of-concept study, the training data has a limited size, training the model with more high quality training data could brought a potential boost. Also, we only used convolutional layers to extract features from NMR spectrum, future work could focus on learning more complicated interaction between peaks as features. As another form of a metric learning, translation models between different NMR spectrum predictors could be desired to better match one prediction to the other.

REFERENCES

[1] A. Filler, “The history, development and impact of computed imaging in neurological diagnosis and neurosurgery: Ct, mri, and dti,” *Nature precedings*, pp. 1–1, 2009.

[2] M. Elyashberg, “Identification and structure elucidation by nmr spectroscopy,” *TrAC Trends in Analytical Chemistry*, vol. 69, pp. 88–97, 2015.

[3] S. L. Robinette, R. Brüschweiler, F. C. Schroeder, and A. S. Edison, “Nmr in metabolomics and natural products research: Two sides of the same coin,” *Accounts of Chemical Research*, vol. 45, no. 2, pp. 288–297, 2012.

[4] D. S. Wishart, “Nmr metabolomics: A look ahead,” *Journal of Magnetic Resonance*, vol. 306, pp. 155–161, 2019.

[5] H. Tsugawa, R. Nakabayashi, T. Mori, Y. Yamada, M. Takahashi, A. Rai, R. Sugiyama, H. Yamamoto, T. Nakaya, M. Yamazaki, *et al.*, “A cheminformatics approach to characterize metabolomes in stable-isotope-labeled organisms,” *Nature methods*, vol. 16, no. 4, pp. 295–298, 2019.

[6] S. Steiner, J. Wolf, S. Glatzel, A. Andreou, J. M. Granda, G. Keenan, T. Hinkley, G. Aragon-Camarasa, P. J. Kitson, D. Angelone, *et al.*, “Organic synthesis in a modular robotic system driven by a chemical programming language,” *Science*, vol. 363, no. 6423, 2019.

[7] L. M. Roch, F. Häse, C. Kreisbeck, T. Tamayo-Mendoza, L. P. Yunker, J. E. Hein, and A. Aspuru-Guzik, “Chemos: Orchestrating autonomous experimentation,” *Science Robotics*, vol. 3, no. 19, 2018.

[8] V. Sans, L. Porwol, V. Dragone, and L. Cronin, “A self optimizing synthetic organic reactor system using real-time in-line nmr spectroscopy,” *Chemical science*, vol. 6, no. 2, pp. 1258–1264, 2015.

[9] R. R. da Silva, P. C. Dorrestein, and R. A. Quinn, “Illuminating the dark matter in metabolomics,” *Proceedings of the National Academy of Sciences*, vol. 112, no. 41, pp. 12 549–12 550, 2015.

[10] K. Ito, Y. Obuchi, E. Chikayama, Y. Date, and J. Kikuchi, “Exploratory machine-learned theoretical chemical shifts can closely predict metabolic mixture signals,” *Chemical science*, vol. 9, no. 43, pp. 8213–8220, 2018.

[11] B. Sanchez-Lengeling and A. Aspuru-Guzik, “Inverse molecular design using machine learning: Generative models for matter engineering,” *Science*, vol. 361, no. 6400, pp. 360–365, 2018.

[12] W. Jin, R. Barzilay, and T. Jaakkola, “Junction tree variational autoencoder for molecular graph generation,” in *International Conference on Machine Learning*, PMLR, 2018, pp. 2323–2332.

[13] J. H. Jensen, “A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space,” *Chemical science*, vol. 10, no. 12, pp. 3567–3572, 2019.

[14] X. Yang, J. Zhang, K. Yoshizoe, K. Terayama, and K. Tsuda, “Chemts: An efficient python library for de novo molecular generation,” *Science and technology of advanced materials*, vol. 18, no. 1, pp. 972–976, 2017.

[15] O. Prykhodko, S. V. Johansson, P.-C. Kotsias, J. Arús-Pous, E. J. Bjerrum, O. Engkvist, and H. Chen, “A de novo molecular generation method using latent vector based generative adversarial network,” *Journal of Cheminformatics*, vol. 11, no. 1, pp. 1–13, 2019.

[16] F. Grisoni, M. Moret, R. Lingwood, and G. Schneider, “Bidirectional molecule generation with recurrent neural networks,” *Journal of chemical information and modeling*, vol. 60, no. 3, pp. 1175–1183, 2020.

[17] Q. Yuan, A. Santana-Bonilla, M. A. Zwijnenburg, and K. E. Jelfs, “Molecular generation targeting desired electronic properties via deep generative models,” *Nanoscale*, vol. 12, no. 12, pp. 6744–6758, 2020.

[18] M. Sumita, X. Yang, S. Ishihara, R. Tamura, and K. Tsuda, “Hunting for organic molecules with artificial intelligence: Molecules optimized for desired excitation energies,” *ACS central science*, vol. 4, no. 9, pp. 1126–1133, 2018.

[19] M. Bühl, M. Kaupp, O. L. Malkina, and V. G. Malkin, “The dft route to nmr chemical shifts,” *Journal of computational chemistry*, vol. 20, no. 1, pp. 91–105, 1999.

[20] J. Aires-de-Sousa, M. C. Hemmer, and J. Gasteiger, “Prediction of 1h nmr chemical shifts using neural networks,” *Analytical Chemistry*, vol. 74, no. 1, pp. 80–90, 2002.

[21] F. Paruzzo, A. Hofstetter, F. Musil, S. De, M. Ceriotti, and L. Emsley, *Chemical shifts in molecular solids by machine learning. nat commun 9 (1): 4501*, 2018.

[22] Y. Binev and J. Aires-de-Sousa, “Structure-based predictions of 1h nmr chemical shifts using feed-forward neural networks,” *Journal of chemical information and computer sciences*, vol. 44, no. 3, pp. 940–945, 2004.

[23] Y. Kwon, D. Lee, Y.-S. Choi, M. Kang, and S. Kang, “Neural message passing for nmr chemical shift prediction,” *Journal of Chemical Information and Modeling*, vol. 60, no. 4, pp. 2024–2030, 2020, PMID: 32250618.

[24] P. Gao, J. Zhang, Y. Sun, and J. Yu, “Toward accurate predictions of atomic properties via quantum mechanics descriptors augmented graph convolutional neural network: Application of this novel approach in nmr chemical shifts predictions,” *The Journal of Physical Chemistry Letters*, vol. 11, no. 22, pp. 9812–9818, 2020, PMID: 33151693.

[25] W. Gerrard, L. A. Bratholm, M. J. Packer, A. J. Mulholland, D. R. Glowacki, and C. P. Butts, “Impression – prediction of nmr parameters for 3-dimensional chemical structures using machine learning with near quantum chemical accuracy,” *Chem. Sci.*, vol. 11, pp. 508–515, 2 2020.

[26] Z. Yang, M. Chakraborty, and A. D. White, “Predicting chemical shifts with graph neural networks,” 2020.

[27] E. Merzbacher, *Quantum mechanics*. Jones & Bartlett Publishers, 1961.

[28] S. A. Hardinger, 2010.

[29] S. Podlewska, W. M. Czarnecki, R. Kafel, and A. J. Bojarski, “Creating the new from the old: Combinatorial libraries generation with machine-learning-based compound structure optimization,” *Journal of Chemical Information and Modeling*, vol. 57, no. 2, pp. 133–147, 2017, PMID: 28158942.

[30] H. Ikebata, K. Hongo, T. Isomura, R. Maezono, and R. Yoshida, “Bayesian molecular design with a chemical language model,” *Journal of computer-aided molecular design*, vol. 31, no. 4, pp. 379–391, 2017.

[31] D. Weininger, “Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules,” *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.

[32] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, “Automatic chemical design using a data-driven continuous representation of molecules,” *ACS central science*, vol. 4, no. 2, pp. 268–276, 2018.

[33] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, “Grammar variational autoencoder,” in *International conference on machine learning*, PMLR, 2017, pp. 1945–1954.

[34] M. H. Segler, T. Kogej, C. Tyrchan, and M. P. Waller, “Generating focused molecule libraries for drug discovery with recurrent neural networks,” *ACS central science*, vol. 4, no. 1, pp. 120–131, 2018.

[35] T. Cazenave and N. Jouandeau, “On the Parallelization of UCT,” in *Computer Games Workshop*, Amsterdam, Netherlands, 2007.

[36] G. M. J. .-B. Chaslot, M. H. M. Winands, and H. J. van den Herik, “Parallel monte-carlo tree search,” in *Computers and Games*, H. J. van den Herik, X. Xu, Z. Ma, and M. H. M. Winands, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 60–71.

[37] X. Yang, “Monte carlo tree search for rna inverse folding and molecule design,” Ph.D. dissertation, Graduate School of Frontier Sciences, The University of Tokyo, Sep. 2018.

[38] X. Yang, T. K. Aasawat, and K. Yoshizoe, “Practical large-scale distributed parallel monte-carlo tree search applied to molecular design,” *ICLR2021*, vol. abs/2006.10504, 2020.

[39] R. G. Parr, “Density functional theory of atoms and molecules,” in *Horizons of quantum chemistry*, Springer, 1980, pp. 5–15.

[40] M. Frisch, G. Trucks, H. Schlegel, G. Scuseria, M. Robb, J. Cheeseman, J. Montgomery Jr, T. Vreven, K. Kudin, J. Burant, *et al.*, *Gaussian 03, revision c. 02*, 2004.

[41] C. Villani, *Optimal transport: old and new*. Springer, 2009, vol. 338.

[42] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000.

[43] P. Brass, *Advanced data structures*. Cambridge University Press Cambridge, 2008, vol. 193.

[44] M. Nakata and T. Shimazaki, “Pubchemqc project: A large-scale first-principles electronic structure database for data-driven chemistry,” *Journal of chemical information and modeling*, vol. 57, no. 6, pp. 1300–1308, 2017.

[45] S. Kuhn and E. Jonas, “Rapid prediction of nmr spectral properties with quantified uncertainty,” 2019.

[46] J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, and A. E. Roitberg, “Approaching coupled cluster accuracy with a

general-purpose neural network potential through transfer learning," *Nature communications*, vol. 10, no. 1, pp. 1–8, 2019.

[47] P. Gao, J. Zhang, Y. Sun, and J. Yu, "Toward accurate predictions of atomic properties via quantum mechanics descriptors augmented graph convolutional neural network: Application of this novel approach in nmr chemical shifts predictions," *The Journal of Physical Chemistry Letters*, vol. 11, no. 22, pp. 9812–9818, 2020.

[48] S. Kang, Y. Kwon, D. Lee, and Y.-S. Choi, "Predictive modeling of nmr chemical shifts without using atomic-level annotations," *Journal of Chemical Information and Modeling*, vol. 60, no. 8, pp. 3765–3769, 2020.

[49] A. Navarro-Vázquez, "State of the art and perspectives in the application of quantum chemical prediction of 1h and 13c chemical shifts and scalar couplings for structural elucidation of organic compounds," *Magnetic Resonance in Chemistry*, vol. 55, no. 1, pp. 29–32, 2017.

[50] M. W. Lodewyk, M. R. Siebert, and D. J. Tantillo, "Computational prediction of 1h and 13c chemical shifts: A useful tool for natural product, mechanistic, and synthetic organic chemistry," *Chemical Reviews*, vol. 112, no. 3, pp. 1839–1862, 2012.

[51] S. Kuhn, B. Egert, S. Neumann, and C. Steinbeck, "Building blocks for automated elucidation of metabolites: Machine learning methods for nmr prediction," *BMC bioinformatics*, vol. 9, no. 1, pp. 1–19, 2008.

[52] S. D. Rychnovsky, "Predicting nmr spectra by computational methods: Structure revision of hexacyclinol," *Organic letters*, vol. 8, no. 13, pp. 2895–2898, 2006.

[53] S. Grimme, C. Bannwarth, S. Dohm, A. Hansen, J. Pisarek, P. Pracht, J. Seibert, and F. Neese, "Fully automated quantum-chemistry-based computation of spin–spin-coupled nuclear magnetic resonance spectra," *Angewandte Chemie International Edition*, vol. 56, no. 46, pp. 14 763–14 769, 2017.

[54] W. Bremser, "Hose—a novel substructure code," *Analytica Chimica Acta*, vol. 103, no. 4, pp. 355–365, 1978.

[55] R. B. Schaller and E. Pretsch, "A computer program for the automatic estimation of 1h nmr chemical shifts," *Analytica chimica acta*, vol. 290, no. 3, pp. 295–302, 1994.

[56] K. A. Blinov, Y. Smurnyy, M. Elyashberg, T. Churanova, M. Kvasha, C. Steinbeck, B. Lefebvre, and A. Williams, "Performance validation of neural network based 13c nmr prediction using a publicly available data source," *Journal of chemical information and modeling*, vol. 48, no. 3, pp. 550–555, 2008.

[57] Y. Binev, M. M. Marques, and J. Aires-de-Sousa, “Prediction of 1h nmr coupling constants with associative neural networks trained for chemical shifts,” *Journal of chemical information and modeling*, vol. 47, no. 6, pp. 2089–2097, 2007.

[58] Y. Binev, M. Corvo, and J. Aires-de-Sousa, “The impact of available experimental data on the prediction of 1h nmr chemical shifts by neural networks,” *Journal of chemical information and computer sciences*, vol. 44, no. 3, pp. 946–949, 2004.

[59] Y. Binev and J. Aires-de-Sousa, “Structure-based predictions of 1h nmr chemical shifts using feed-forward neural networks,” *Journal of chemical information and computer sciences*, vol. 44, no. 3, pp. 940–945, 2004.

[60] J. Meiler, R. Meusinger, and M. Will, “Fast determination of 13c nmr chemical shifts using artificial neural networks,” *Journal of chemical information and computer sciences*, vol. 40, no. 5, pp. 1169–1176, 2000.

[61] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *International Conference on Machine Learning*, PMLR, 2017, pp. 1263–1272.

[62] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” *IEEE transactions on neural networks and learning systems*, 2020.

[63] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, “Relational inductive biases, deep learning, and graph networks,” *arXiv preprint arXiv:1806.01261*, 2018.

[64] Z. Yang, M. Chakraborty, and A. D. White, “Predicting chemical shifts with graph neural networks,” 2020.

[65] S. J. Prestrelski, N. Tedeschi, T. Arakawa, and J. F. Carpenter, “Dehydration-induced conformational transitions in proteins and their inhibition by stabilizers,” *Biophysical journal*, vol. 65, no. 2, pp. 661–671, 1993.

[66] J. F. Carpenter, S. J. Prestrelski, and T. Arakawa, “Separation of freezing-and drying-induced denaturation of lyophilized proteins using stress-specific stabilization: I. enzyme activity and calorimetric studies,” *Archives of Biochemistry and Biophysics*, vol. 303, no. 2, pp. 456–464, 1993.

[67] B. S. Kendrick, A. Dong, S. D. Allison, M. C. Manning, and J. F. Carpenter, “Quantitation of the area of overlap between second-derivative amide i infrared spectra to determine the structural similarity of a protein in different states,” *Journal of pharmaceutical sciences*, vol. 85, no. 2, pp. 155–158, 1996.

[68] L. Bodis, A. Ross, and E. Pretsch, “A novel spectra similarity measure,” *Chemometrics and Intelligent Laboratory Systems*, vol. 85, no. 1, pp. 1–8, 2007.

[69] A. M. Castillo, L. Uribe, L. Patiny, and J. Wist, “Fast and shift-insensitive similarity comparisons of nmr using a tree-representation of spectra,” *Chemometrics and Intelligent Laboratory Systems*, vol. 127, pp. 1–6, 2013.

[70] M. Kaya and H. Ş. Bilge, “Deep metric learning: A survey,” *Symmetry*, vol. 11, no. 9, p. 1066, 2019.

[71] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *International workshop on similarity-based pattern recognition*, Springer, 2015, pp. 84–92.

[72] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[73] A. M. Castillo, A. Bernal, L. Patiny, and J. Wist, “A new method for the comparison of 1 h nmr predictors based on tree-similarity of spectra,” *Journal of cheminformatics*, vol. 6, no. 1, pp. 1–6, 2014.

[74] R. J. Abraham and M. Mobli, “The prediction of 1h nmr chemical shifts in organic compounds,” *Spectroscopy Europe*, vol. 16, no. 4, pp. 16–22, 2004.

[75] O.-E. Ganea, L. Pattanaik, C. W. Coley, R. Barzilay, K. F. Jensen, W. H. Green, and T. S. Jaakkola, “Geomol: Torsional geometric generation of molecular 3d conformer ensembles,” *arXiv preprint arXiv:2106.07802*, 2021.

[76] K. Ito, Y. Tsutsumi, Y. Date, and J. Kikuchi, “Fragment assembly approach based on graph/network theory with quantum chemistry verifications for assigning multidimensional nmr signals in metabolite mixtures,” *ACS chemical biology*, vol. 11, no. 4, pp. 1030–1038, 2016.

Appendices

APPENDIX A

PSEUDO-CODE FOR TRIE ENHANCEMENT

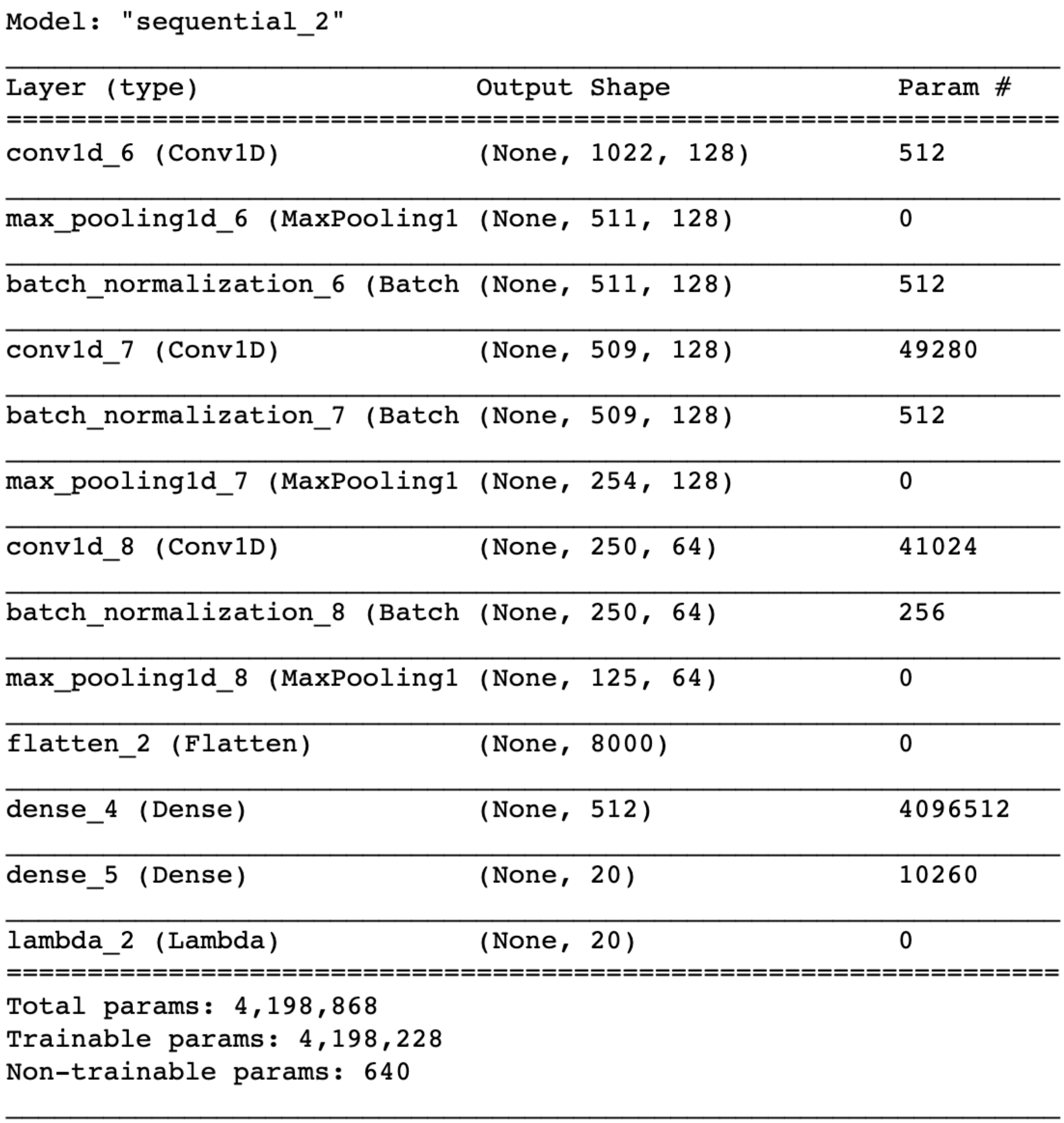
```
if $trieSize > 0$ then
    /* Read NMR Database                                          */
    $NMRList, SMILESList \leftarrow readTrieDatabase()$;
    for $i \leftarrow 0$ to $len(NMRList) - 1$ do
        $WSList[i] \leftarrow getWassersteinScore(NMRList[i], targetNMR)$;
        /* Note:  NMRList and targetNMR contains element
           information of each peak(atom), the atom number
           penalty can therefore be computed                      */
    end
    /* Sort and select top-N data points depends on
       trieSize                                                   */
    $Data \leftarrow rowStack(SMILESList, WSList, NMRList)$;
    $sortedData \leftarrow sort(data, key = WSList)$;
    $SMILESList \leftarrow sortedData[: trieSize, 0]$;
end
```

```
/* Create Trie                                                  */
for $i \leftarrow 0$ to $len(SMILESList) - 1$ do
    $j \leftarrow 0$;
    $state \leftarrow ['\&']$;
    $currentNode \leftarrow rootNode$;
    $nodesList \leftarrow SMILESToNodes(smiles[i])$;
    /* Map database SMILES into prefix search tree              */
    $k \leftarrow 0$;
    while $k <$ len(nodesList) do
        $state[k] \leftarrow nodesList[k]$;
        if nodesList[k] currentNode.childNodes then
            $newNode \leftarrow createNewChildNode(position =$
              $nodesList[k], parentNode = currentNode)$;
            $currentNode \leftarrow newNode$;
        else
            for $child\ \ currentNode.childNodes$ do
                if $child.position == nodesList[k]$ then
                    $currentNode \leftarrow child$;
                    $break$;
                end
            end
        end
        $k \leftarrow k + 1$;
        /* Update traversed nodes with the Wasserstein
           Score                                                */
        $currentWS \leftarrow getWassersteinScore(NMRList[k], targetNMR)$;
        while $currentNode! = None$ do
            $currentNode.Update(currentWS)$;
            $currentNode \leftarrow currentNode.parentNode$;
        end
    end
end
```

APPENDIX B

SIAMESE NEURAL NETWORK STRUCTURE

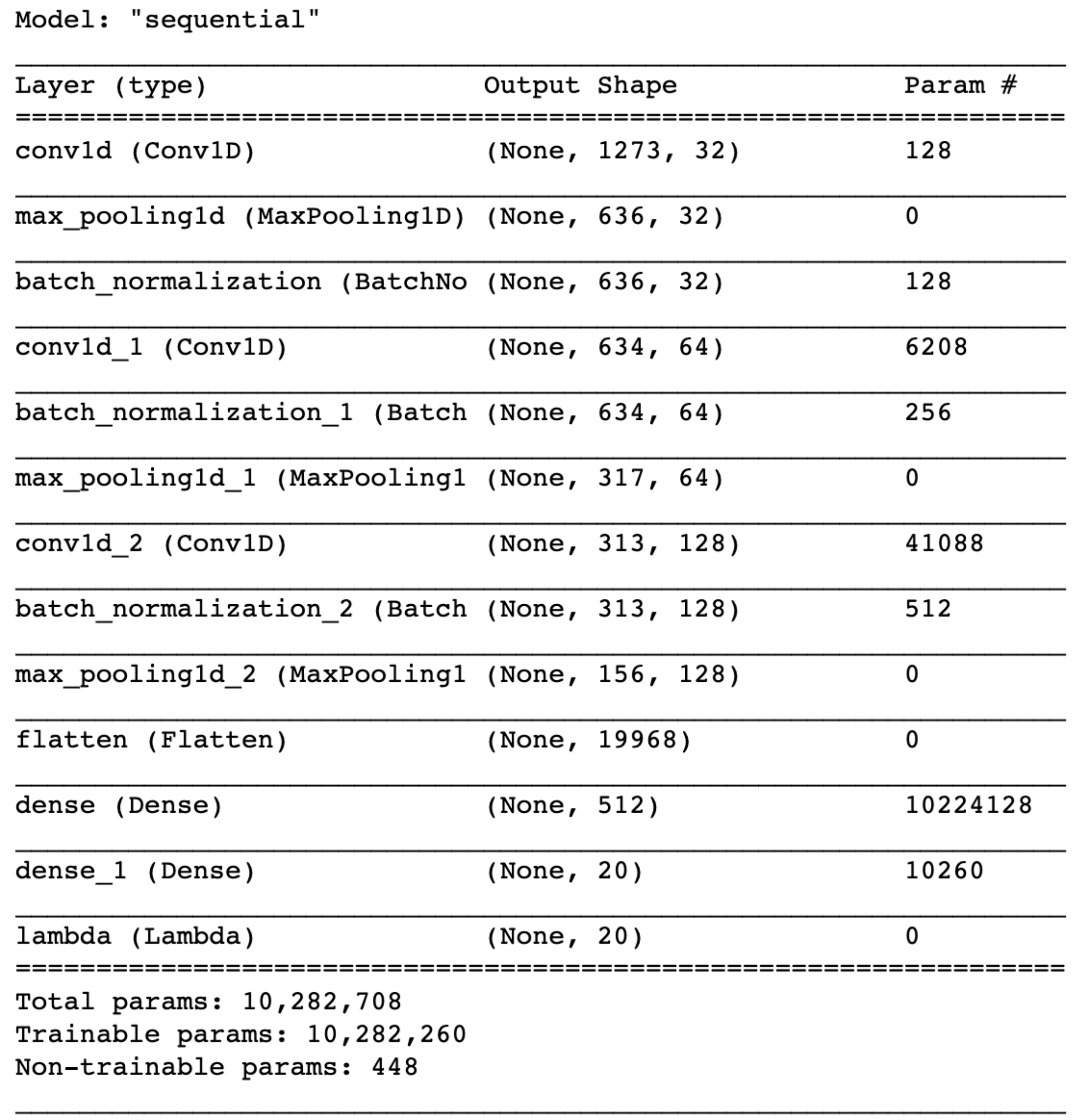
B.1 Neural network structure for Intensity Representation input



```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_6 (Conv1D)            (None, 1022, 128)         512
_________________________________________________________________
max_pooling1d_6 (MaxPooling1 (None, 511, 128)          0
_________________________________________________________________
batch_normalization_6 (Batch (None, 511, 128)          512
_________________________________________________________________
conv1d_7 (Conv1D)            (None, 509, 128)          49280
_________________________________________________________________
batch_normalization_7 (Batch (None, 509, 128)          512
_________________________________________________________________
max_pooling1d_7 (MaxPooling1 (None, 254, 128)          0
_________________________________________________________________
conv1d_8 (Conv1D)            (None, 250, 64)           41024
_________________________________________________________________
batch_normalization_8 (Batch (None, 250, 64)           256
_________________________________________________________________
max_pooling1d_8 (MaxPooling1 (None, 125, 64)           0
_________________________________________________________________
flatten_2 (Flatten)          (None, 8000)              0
_________________________________________________________________
dense_4 (Dense)              (None, 512)               4096512
_________________________________________________________________
dense_5 (Dense)              (None, 20)                10260
_________________________________________________________________
lambda_2 (Lambda)            (None, 20)                0
=================================================================
Total params: 4,198,868
Trainable params: 4,198,228
Non-trainable params: 640
_________________________________________________________________
```

Figure B.1: Detailed siamese neural network structure Intensity Representation input

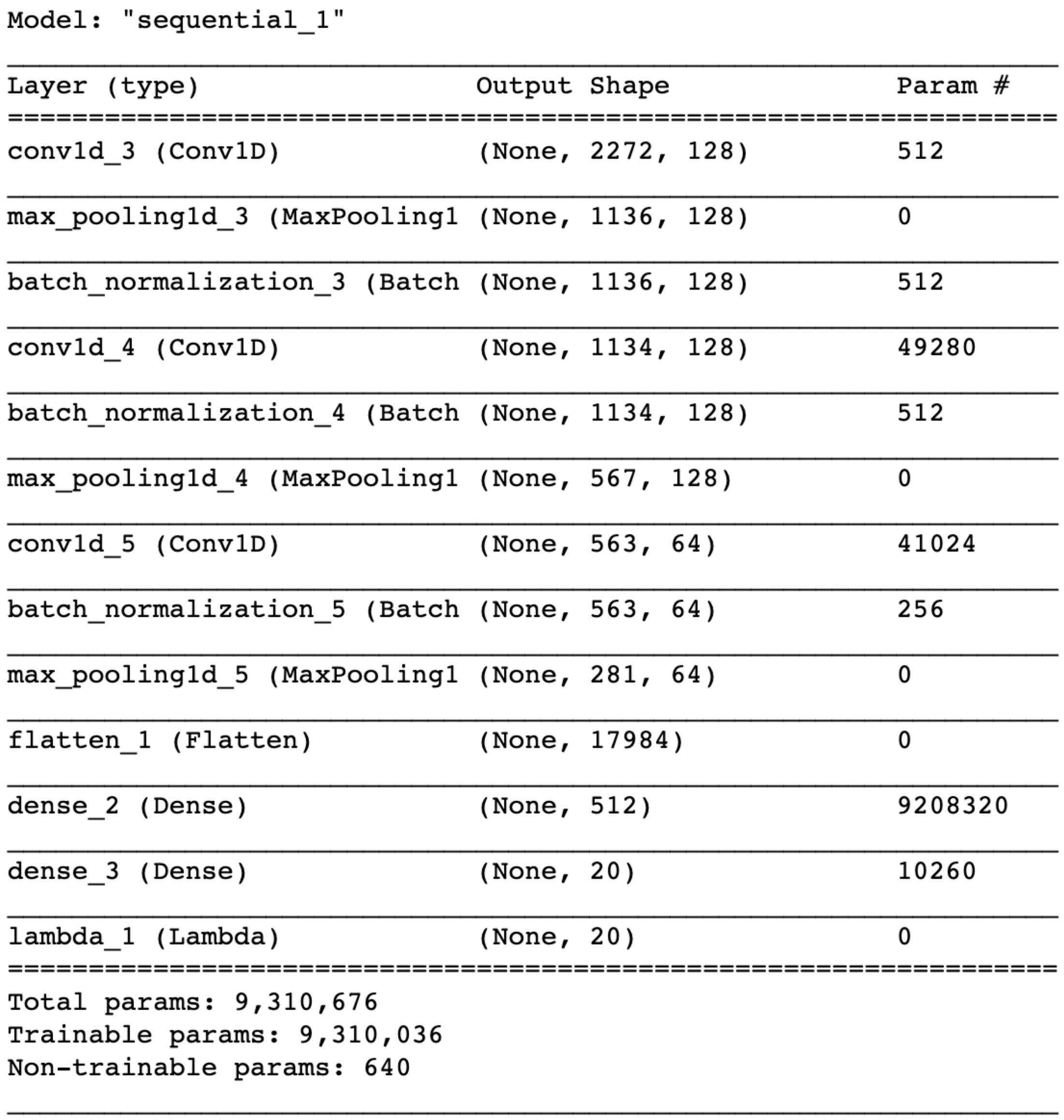
B.2 Neural network structure for Bin Representation input



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1d (Conv1D)              (None, 1273, 32)          128       
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 636, 32)           0         
_________________________________________________________________
batch_normalization (BatchNo (None, 636, 32)           128       
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 634, 64)           6208      
_________________________________________________________________
batch_normalization_1 (Batch (None, 634, 64)           256       
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 317, 64)           0         
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 313, 128)          41088     
_________________________________________________________________
batch_normalization_2 (Batch (None, 313, 128)          512       
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 156, 128)          0         
_________________________________________________________________
flatten (Flatten)            (None, 19968)             0         
_________________________________________________________________
dense (Dense)                (None, 512)               10224128  
_________________________________________________________________
dense_1 (Dense)              (None, 20)                10260     
_________________________________________________________________
lambda (Lambda)              (None, 20)                0         
=================================================================
Total params: 10,282,708
Trainable params: 10,282,260
Non-trainable params: 448
_________________________________________________________________
```

Figure B.2: Detailed siamese neural network structure for Bin Representation input

B.3 Neural network structure for Intensity + Bin Representation input



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1d_3 (Conv1D)            (None, 2272, 128)         512       
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 1136, 128)         0         
_________________________________________________________________
batch_normalization_3 (Batch (None, 1136, 128)         512       
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 1134, 128)         49280     
_________________________________________________________________
batch_normalization_4 (Batch (None, 1134, 128)         512       
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 567, 128)          0         
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 563, 64)           41024     
_________________________________________________________________
batch_normalization_5 (Batch (None, 563, 64)           256       
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 281, 64)           0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 17984)             0         
_________________________________________________________________
dense_2 (Dense)              (None, 512)               9208320   
_________________________________________________________________
dense_3 (Dense)              (None, 20)                10260     
_________________________________________________________________
lambda_1 (Lambda)            (None, 20)                0         
=================================================================
Total params: 9,310,676
Trainable params: 9,310,036
Non-trainable params: 640
_________________________________________________________________
```

Figure B.3: Detailed siamese neural network structure for Intensity + Bin Representation input