

Spectra to Structure: Deep Reinforcement Learning for Molecular Inverse Problem

Bhuvanesh Sridharan¹, Sarvesh Mehta¹, Yashaswi Pathak¹,
U Deva Priyakumar^{1,†}

¹International Institute of Information Technology, Hyderabad

† deva@iiit.ac.in

Abstract

Spectroscopy is the study of how matter interacts with electromagnetic radiations of specific frequencies that has led to several monumental discoveries in science. The spectra of any particular molecule is highly information-rich, yet the inverse relation from the spectra to the molecular structure is still an unsolved problem. Nuclear Magnetic Resonance (NMR) spectroscopy is one such critical tool in the tool-set for scientists to characterise any chemical sample. In this work, a novel framework is proposed that attempts to solve this inverse problem by navigating the chemical space to find the correct structure that resulted in the target spectra. The proposed framework uses a combination of online Monte-Carlo-Tree-Search (MCTS) and a set of offline trained Graph Convolution Networks to build a molecule iteratively from scratch. Our method is able to predict the correct structure of the molecule $\sim 80\%$ of the time in its top 3 guesses. We believe that the proposed framework is a significant step in solving the inverse design problem of NMR spectra to molecule.

Introduction

Spectroscopy is the study of interactions between matter and electromagnetic waves. This study has paved the way for many monumental discoveries and theories in science, ranging from quantum mechanics to chemical sciences. Spectroscopy in general has played a significant role in diverse applications such as drug discovery, protein structure determination and material discovery. Nuclear Magnetic Resonance (NMR) spectroscopy is one of the most crucial and versatile methods for chemists and scientists to perform chemical characterization. It is an analytical technique based on the nuclei's magnetic properties that either have an odd mass number or an even mass number with an odd atomic number. Nuclei with non-zero spin \vec{S} would always have a non-zero magnetic dipole moment, $\vec{\mu}$. NMR relies on this for the nuclei to respond to perturbations in the presence of an external magnetic field. NMR spectroscopy is a critical and often the only method to obtain high-resolution information about proteins, DNA, and RNA (Asami, Schmieder, and Reif 2010) (Spronk et al. 2002). It can also be used to obtain knowledge of energy minima and barriers by observing conformational dynamics of proteins (Balazs et al. 2021). This can be pivotal in the process of drug discovery.

The ^{13}C NMR spectra measures the properties of individual nuclei and consists of peaks that correspond to each car-

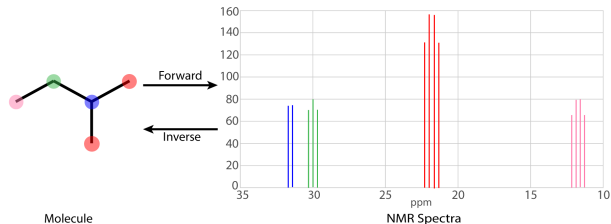


Figure 1: 2-methyl butane and its NMR spectra. Each carbon and its corresponding peak is indicated by the same colour.

bon atom present in the molecule. The peak position (chemical shifts) and the peak splits (spin-spin coupling) are dependent on the local environment of that atom. Usually in labs, experts manually identify the molecular structure from the NMR spectra using highly specific domain knowledge. Till today, most computer-based methods to verify the structure of a sample from its NMR spectra rely on matching the spectral data with a database of already known spectra (Koichi et al. 2014) (Vliegthart, van Kuik, and Hård 1992). These methods restrict the usage to identifying only those molecules that are stored in the database.

The problem concerned here is a non-linear inverse problem. Forward model $y = f(x)$, in this context, refers to the task of calculating the NMR spectra y , given a molecule x . Whereas, the inverse problem refers to drawing conclusions about an unobserved molecule x from its experimentally observed NMR spectra y (Figure 1).

The forward problem f for NMR spectra is relatively well studied with many methods ranging from quantum mechanical calculations and density functional theory (Lodewyk, Siebert, and Tantillo 2012) to deep learning (Jonas and Kuhn 2019) to solve the task. Other empirical methods such as featurizing the neighbourhood of a nuclei and then matching it against a database of known motifs to predict its shift are also common (Bremser 1978) (Cherinka et al. 2019) (Mills 2006).

Recently, there have been many significant studies on the use of modern deep learning and RL methods to solve problems in chemical sciences ranging from prediction of properties of molecules to de novo molecule generation with optimized properties (Zhou et al. 2018) (You et al. 2018). There are var-

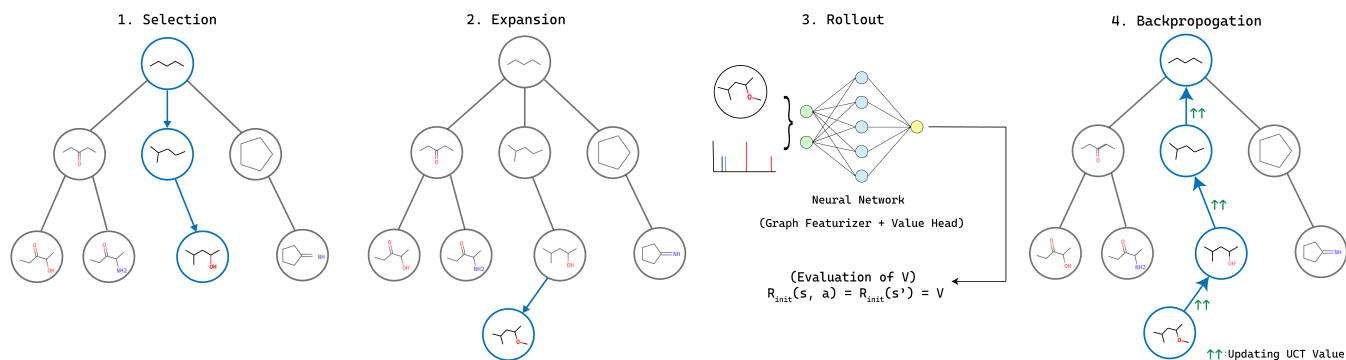


Figure 2: Monte Carlo Tree Search: A heuristic search algorithm where each node in tree is a state of the environment. One of these 4 steps is taken at each step to navigate the search space. **1. Selection:** New bonds are added to the root based on UCT values until a leaf node is reached, **2. Expansion:** A new node is added to the leaf node after environment checks the validity of new node. **3. Rollout:** Value neural network evaluates the value of the newly added node. **4. Back-propagation:** The tree then back-propagates the new information to update the UCT values of all the nodes till the root node.

ious high-throughput combinatorial methods to generate organic molecules with desired properties that are well known and essential in the process of drug discovery (Liu, Li, and Lam 2017) (Benz et al. 2019). In such a workflow, it would be of great help to have a framework to verify the structures of samples generated *in situ* based on easily acquirable spectral data.

In this work, efforts have been made to determine the structure of a molecule, given its NMR spectra and molecular formula. There have been a few endeavours to solve this inverse problem. (Zhang et al. 2020) used a tree-based search framework with a SMILES Generator to predict the structure from computationally generated ^1H NMR spectra. Their method included help from computationally expensive DFT Calculations to guide the tree and were able to predict structure from six out of nine given spectra. In a work by (Jonas 2019), a graph neural network is trained on molecular graphs with imitation learning. The NMR spectra is incorporated as per-node information in the molecular graph, and the molecule is built iteratively by adding edges based on the probabilities returned by the neural network.

In this work, we use a combination of online Monte Carlo Tree Search (MCTS) (Silver et al. 2017) and a set of offline trained Graph Convolutional Networks (Gilmer et al. 2017) to navigate through the chemical space and find the correct molecular structure of a given target ^{13}C NMR spectra.

Dataset

We use nmshiftdb2 (Kuhn and Schlörner 2015) which is a database for organic structures and their experimentally calculated NMR spectra. In this work, we only consider organic molecules that have less than 10 non-hydrogen atoms (C, O, N, F). Charged molecules and radicals are also excluded. Thus, the dataset comprises a total of 2134 molecules with experimentally obtained chemical shift and split values of ^{13}C NMR spectra.

Method

To reiterate, the problem is defined in the following way:

- We have ^{13}C NMR spectra of the target molecule consisting of each carbon’s shift and split values and the molecule’s molecular formula.
- The process of solving the problem is modelled as a Markov Decision Process wherein the molecule is built iteratively from scratch by adding atoms and bonds to the current structure at each step.

State Representation

- The current state in the search process is represented as a **Molecular Graph**.
- Each atom in the target molecule is present in the current state as a node. The graph of the current state has $n - s + 1$ components, where n is the total number of atoms in the target molecule, and s is the number of atoms present in molecule of the current state. Out of these $n - s + 1$ islands, one is a connected component representing the molecule of the current state, and the rest of the $n - s$ islands are individual atoms that may join the current molecule by addition of new bonds later on.
- **Featurization of the target NMR:** Each NMR peak is assigned to a carbon in the beginning when the state consists only of individual nodes and no edges. The node feature of an atom consists of the one-hot encoding of the element kind and the current implicit valency of that atom. A Gaussian, with the peak of the assigned shift centred at the chemical shift value and $\sigma = 2$, is discretized into 64 bins. This feature is then appended to the node feature.

Action Representation

In our work, we formulate a fixed-dimension action space in which each action signifies the addition of an edge between any two nodes in the graph. The environment ensures the validity of these actions by checking for the following conditions:

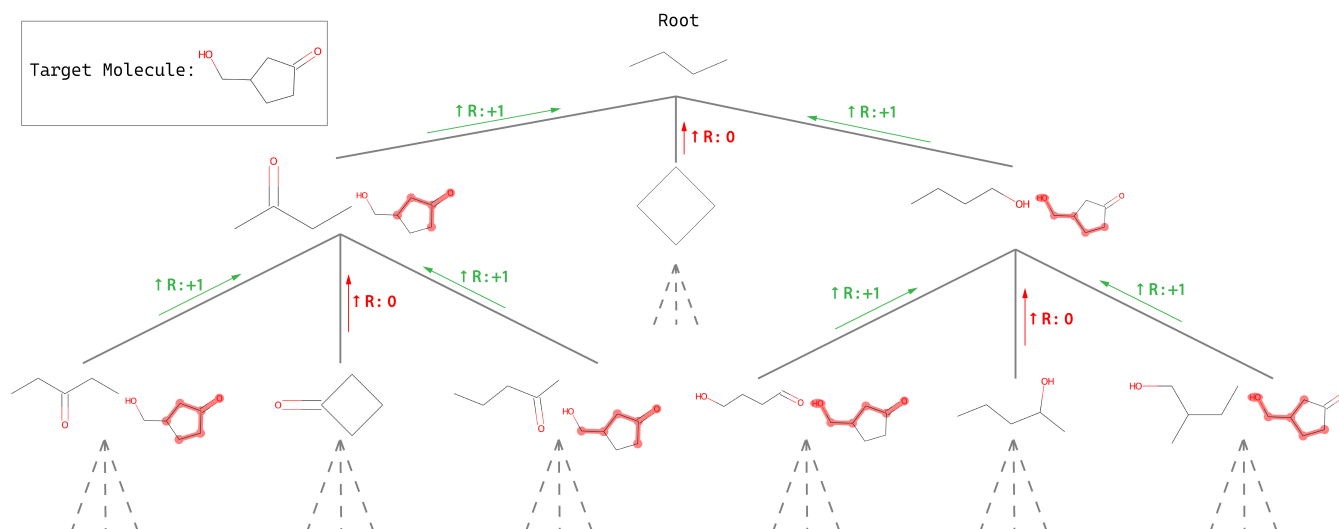


Figure 3: Training Methodology : An example search tree while in the training mode on how subgraph isomorphism is used to make dataset for the neural networks to train on. The target molecule is shown on the top left . Assuming that the current state in the environment is n-butane, we see a possible state of the search tree. Each node in the tree is also accompanied by an illustration showing how it is a subgraph of the target molecule. Since this can be evaluated when the training mode is on, this is used to return intermediate reward $r(s, a) = 1$ when the current state is subgraph isomorphic to the target state, otherwise $r(s, a) = 0$.

- At least one of the endpoints of the newly added edge must belong to the sub-graph containing molecule of the current state.
- The addition of this new edge must obey the chemical rules of valency for each atom. If the valency due to connection with other heavy atoms is not enough to complete its octet, it is implicitly assumed that the rest of the valency is satisfied by hydrogens. These hydrogens are not taken as nodes in the molecular graph.
- The edge must not lead to formation of a ring with four or three atoms.
- The edge must not lead to a bond within an already present ring.¹

Agent

Since we have formulated the task at hand as a Markov Decision Process, we are left to decide on a planning algorithm that would use some prior knowledge about the problem and explore various branches of the search tree before taking action a on a state s . One such algorithm is Monte Carlo Tree Search (MCTS) (Figure 2). MCTS performs one of the four following steps repeatedly:

1. **Select** : In this stage of MCTS, the tree is traversed from the root according to the UCT (Upper Confidence Bound for Trees) values at each level until it reaches a leaf

¹Note that this restriction does not prevent the formation of bicyclo and spiro compounds. It just guides the formation so that the smaller ring is formed before the larger ring. Doing so proved to be helpful in the initial experiments since this helps prune some redundant branches of the tree search.

node. The UCT value at any state is calculated based on the following formula:

$$UCT(s, a) = Q(s, a) + c * \pi_{\text{model}}(a|s) * \frac{\sqrt{N(s) + 1}}{n(s, a) + 1}$$

Where s : Current state, $Q(s, a)$: Mean Action Value estimate $Q(s, a) = \frac{W(s, a)}{n(s, a)}$, $W(s, a)$: the cumulative of all returns $R(s', a')$ till the leaf node, $\pi_{\text{model}}(a|s)$: Prior probability by the policy network, $N(s)$: Number of times state s has been reached, $n(s, a)$ is the number of times action a was taken from state s , and c is the constant with which one can manipulate exploration vs exploitation ratio. The form of UCT value used in this work is inspired by (Moerland et al. 2018), which proved to improve the performance of cases with asymmetric trees.

2. **Expand**: Once a leaf node s'_L is reached by the tree search, the tree is expanded by addition of a new leaf node s_L . The environment simulates this action and ensures its validity and also returns an intermediate reward.

3. **Roll-out**: In a typical MCTS, the initial value of the new leaf node is estimated using a series of random rollouts from the leaf node s_L . Due to computational limitations, this work uses a value neural network $V_{\text{model}}(s)$ to estimate the value function.

4. **Back propagation**: After estimating the value of the newly added leaf node, $R(s, a)$ of the whole backward trace is updated through back propagation which in turn updates the UCT value of intermediate nodes belonging to this trace.

$$R(s_i, a_i) = r(s_i, a_i) + \gamma R(s_{i+1}, a_{i+1})$$

The above four steps are repeated for $nmcts$ number of times. Then, a real action a_t is taken by the environment based on the policy of the tree. The tree’s policy probability is determined by the visitation count of all the actions at the root node s_0 .

Role of the Forward Predictor

While the inverse problem is defined as the task to determine the molecular structure from the spectra, it naturally follows that the forward problem is that of calculating the NMR spectra given the molecule and its structure. Here, a forward NMR prediction model (Jonas and Kuhn 2019) is used for the following:

1. **For intermediate reward:** Typical MCTS applications also have an intermediate reward returned by the environment for each action. The step reward is calculated based on how close is the current state to the target molecule. The forward model predicts the NMR spectra of the current state and the reward is defined by:

$$r(s, a) = r(s') = 2 * (\frac{1}{2} - WS(s'))$$

$$WS(s') = \text{First Wasserstein Distance} = l_1(S_T, S_C)$$

$$l_1(u, v) = \int_{-\infty}^{\infty} |U - V|$$

where U and V are the CDFs for the distribution of some random variables u and v (Ramdas, Garcia, and Cuturi 2015), S_T is the NMR spectra of the target molecule, and S_C is the NMR spectra of the current molecules'. The reward $r(s') = WS(s')$ is returned whenever the current state s' is known to be a terminal state. Otherwise, $r(s') = 0$ is returned.

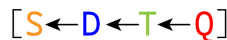
2. **For the Scoring Function:** Each episode performed by the agent returns one prediction of what the target molecule is. Since MCTS has some element of randomness, all guesses made by the agent are not the same. In such cases, after running the agent for a predetermined fixed number of times, all the unique guesses are ranked against each other by the means of the reward function discussed above. Then, the guess which returns the highest reward is taken as the final prediction.

Architecture of the Prior Policy and Value network

There are three modules of neural network used in this work.

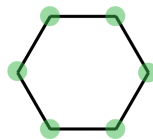
Graph Featurizer: This module uses Message Passing Neural Network (Gilmer et al. 2017) (Duvenaud et al. 2015), which provides a formulation for supervised learning on graph structured data. Consider a **molecular graph** $G(V, E)$ with node features x_v (having information about the current state and also the target NMR spectra) and edge features e_{vw} . The features of each node at time step t are represented as h_v^t , initialized to x_v at $t = 0$. The features of nodes are updated for 3 time steps using messages m_v^{t+1} in the following way:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$



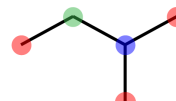
In the state, the carbons can transfer their split category only unidirectionally from right to left.

a) Target



[0, 0, 6, 0]

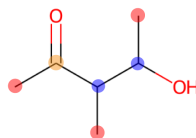
Current State



TERMINAL: True

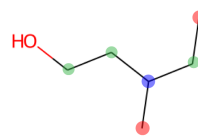
[0, 1, 1, 3]

b) Target



[1, 2, 0, 3]

Current State



TERMINAL: True

[0, 1, 3, 2]

Figure 4: a) Target state of cyclohexane and current state of 2-methyl-butane along with their splitvectors b) Target state of 4-hydroxy-3-methylpentan-2-one and current state of 3-methylpentan-1-ol along with their splitvectors

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

$$F_v = g(x_v, h_v^t) = x_v + h_v^t, \forall v \in V$$

where $N(v)$ is the set of neighbouring nodes of v . M_t and U_t are the message function and vertex update function respectively. The function g is simply taken to be vector addition in this work. F_v is the final atomic feature for the node which has information about the atomic properties, the local environment, and also the target NMR shift value that was assigned to this node. The feature (F_v) generated here will be further used by the policy neural network $\pi_{\text{model}}(a|s)$ and value network $V_{\text{model}}(s)$.

Policy Head: N nodes form $\binom{N}{2}$ pairs, each representing a possible edge. For each of these pairs, let the feature vector of the pair be the concatenation of the feature vector of the two nodes concerned. This pair’s feature vector is then passed through two fully connected layers to obtain a 3-tuple representing the possibility of single, double, and triple bond between this pair.

Value Head: All the node features received from the graph featurizer are then sum-pooled to attain a molecule-level feature vector which has information about both the current molecule and the target NMR. This molecule level feature is then passed through two fully connected layers to finally predict the value $V_{\text{model}}(s)$ of the current state.

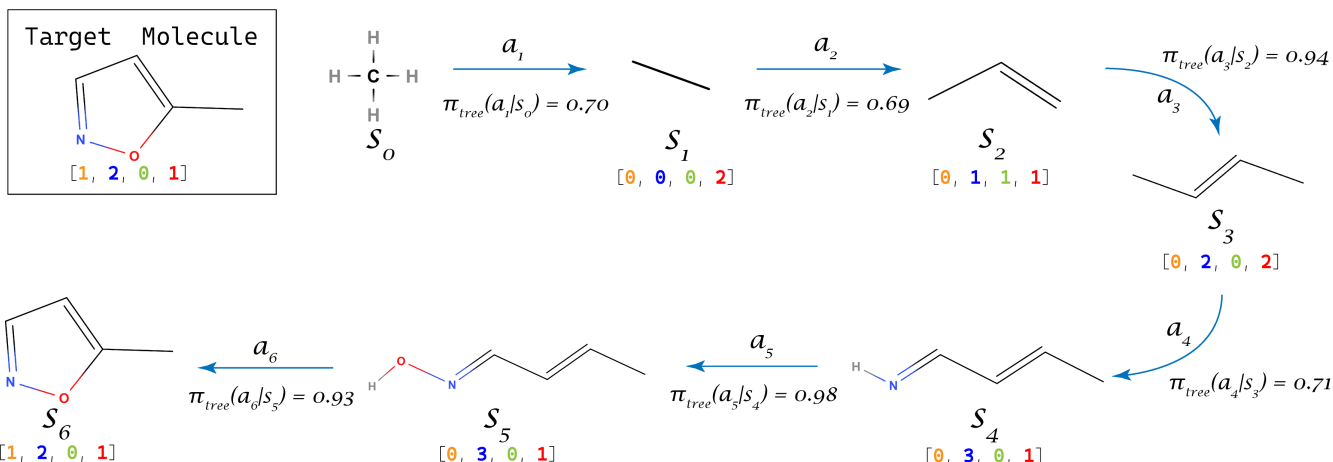


Figure 5: An example run for the target molecule CC1=CC=NO1 with $nmcts = 1000$, $\pi_{\text{tree}}(a_i|s_j)$ represents the probability of taking action a_i according to the policy returned by the MCTS Search with state s_j as the root. In the above figure, each state s_j is also accompanied by the splitvector of that state.

Training Methodology

While in the training mode, the environment has access to not only the NMR spectra but also the structure of the target molecule. This can be used to guide the tree by giving a strong positive reinforcement in the form of $r(s, a)$ (Figure 3). The tree policy (derived from visitation counts of the actions) and approximation of $Q(s, a)$ hence obtained is used as the training dataset for the prior policy neural network $\pi_{\text{model}}(a|s)$ and value network $V_{\text{model}}(s)$. When any action a at state s leads it to state s' , i.e $s \xrightarrow{a} s'$, then:

$$r(s, a) := 1 \quad \text{iff } S(s_t, s'), \quad \text{else } 0$$

where $S(s_t, s')$ is Boolean function that returns True iff s' is subgraph isomorphic to the s_t . This work employs rdkit(Landrum 2016) to check whether the state s' is a subgraph of the target molecule s_t . With training mode on, the model was run on system with Intel Xeon E5-2640 v4 processor and Nvidia GeForce GTX 1080 Ti GPU for 23 Hours to collect experience and train the neural networks. 5 models were trained on different crossvalidation training sets.

Using the Split Information to prune the trees

Each shift value in the dataset is accompanied by a split value as well. The split value is a categorical variable that belongs to one of $\{S, D, T, Q\}$, and it is dependent on the number of hydrogen atoms that are attached to the carbon. A quaternary carbon (no hydrogen attached) leads to a singlet (S) split, a tertiary carbon (one hydrogen attached) leads to a doublet (D) split, a secondary carbon (two hydrogens attached) leads to a triplet (T) split, and a primary carbon (three hydrogens) leads to a quartet (Q) split. Let splitvector be the vector that stores the information about the number of carbons of each split kind in the current state. Since the only action possible in the modelled MDP is that of addition of an edge(decreases the number of implicit hydrogens), note the following two invariant properties:

- The sum of values in the splitvector would remain constant for states with only 1 connected component since the total number of carbons can't increase.
- With addition of bonds, the kind of split made by a particular carbon can only move in the direction:

$$S \leftarrow D \leftarrow T \leftarrow Q$$

As a consequence of this, certain states can be flagged as terminal states if it is known that they can never lead to the target molecule based on the following criteria:

- When the number of quaternary carbons in the current state becomes lower than the number of quartet splits in the target spectra.
- When the number of singlet carbons in the current state becomes more than the number of singlet splits in the target spectra and so on.

For example, in Figure 4 a, the agent can safely terminate search through this branch since once a duplet has formed in the current state, that carbon can never be transformed back to triplet or quartet and we know that the target molecule does not have any duplet or singlet carbon. Similarly, in Figure 4 b, the agent can safely terminate since the number of quartets in the current state has gone below the number of quartets in the target molecule and there is no way to produce new quartet carbon atoms. These chemistry guided conditions greatly prune the search tree and prevents the tree from exploring branches that can lead to the incorrect structure.

Results and Discussion

Accuracy of the Forward model

The forward model used in this work (Jonas and Kuhn 2019) was trained on nmrshiftdb2 dataset (Kuhn and Schlörner 2015) as included in the original work. The mean absolute error obtained for the prediction of the shiftvalue per peak for the predictor was 1.374 ppm.

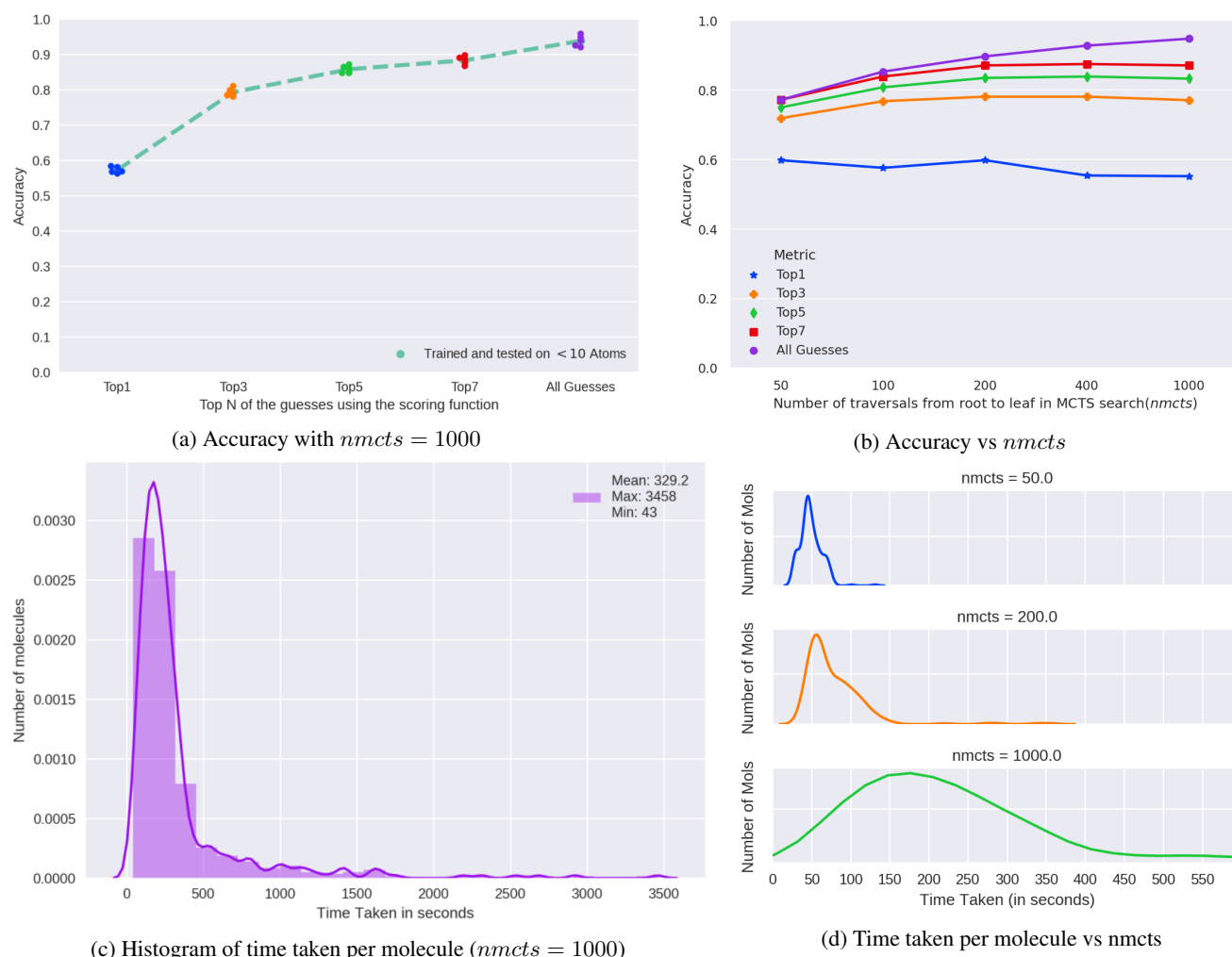


Figure 6: a) Accuracy over 5-fold cross-validation with $nmcts = 1000$ b) Effect of $nmcts$ (Number of traversal from the root to leaf in MCTS search) on the various metrics of accuracy, c) Histogram of time taken for the model to run on each molecule for $nmcts = 1000$, d) Effect of $nmcts$ on the time taken by the system to predict a molecule

Crossvalidation

The total dataset of 2134 molecules was randomly split into 5 equal groups. In each of the five experiments, one of the groups was chosen as the hold-out test dataset and the model was trained on the remaining four groups. For each molecule, the agent made a number of guesses depending on how many episodes it ran. There were 20 processes initiated, with each of them running an episode. An example of one of such episode runs is illustrated in Figure 5. As shown in Figure 6 a, on an average, the agents guessed the correct structure of the molecule of the target spectra 93.8% of the time. All the guesses of the agent are then ranked based on the scoring function discussed in the earlier section. The Top1 ranked structure among the guesses was the target structure 57.2% of the time. Accuracy for Top3, Top5, and Top7 of the scored guesses can be seen in Figure 6 a.

Effect of $nmcts$ on the Accuracy

$nmcts$ is the number of times that the search tree is traversed from the root to leaf node to explore different branches before making a true action in the current state. As expected, it can be observed in Figure 6 b that the net accuracy improves as $nmcts$ increases.

It is also seen that the trend for Top1 accuracy is not the same as others and it actually decreases with increase in $nmcts$. This can be reasoned with the fact the increasing $nmcts$ increases the exploration of the chemical space and more potential candidates are scored against the current structure. This downward trend reveals that a better scoring function would improve the TopN accuracy of the agent since it would be able to rank the candidate guesses in a more accurate way.

Figure 6 d shows the distribution of time taken for each molecule with varying $nmcts$. As expected, we observe an increase in time taken for the agent to make all its guesses

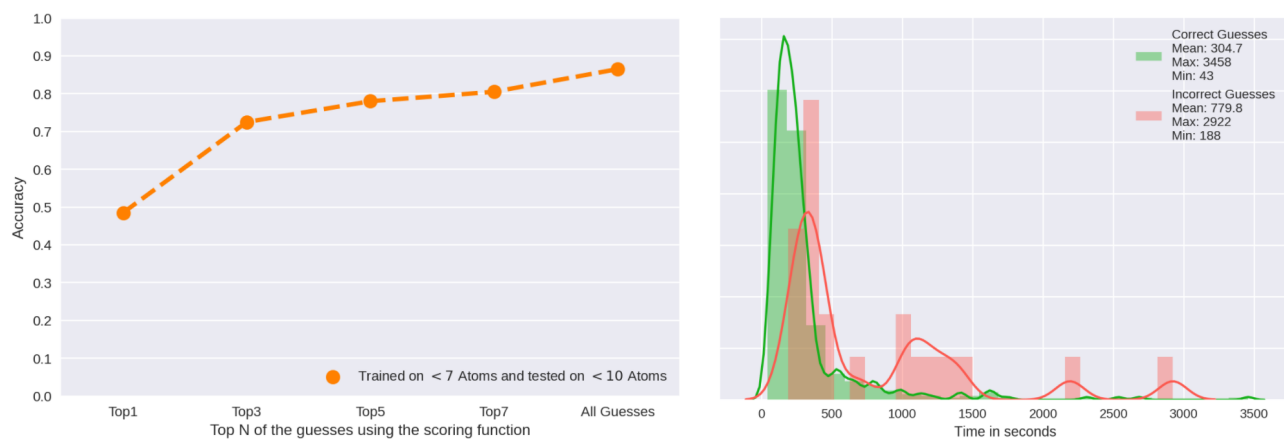


Figure 7: a) Accuracy when the model is trained of molecules with < 7 atoms and tested on molecules with < 10 atoms. b) Time taken for a molecule when it is guessed correctly and incorrectly ($nmcts = 1000$)

for a target NMR spectra.

Holdout experiment with training data

In another experiment, molecules with < 7 non-hydrogen atoms were filtered from the dataset. After running the agent on these filtered set of molecules with training mode on, the agent was tested on 200 randomly sampled molecules with ≥ 7 and < 10 heavy atoms. The result of this experiment is plotted in the Figure 7 a. However, the system performs well on the class of data that it was never exposed to before, by guessing the correct structure 86.5% of the time.

Time difference between correctly guessed and incorrectly guessed molecules

The histogram of the time taken for the agent to run all the episodes for a molecule can be seen in Figure 6 c. On an average, it takes ≈ 330 seconds for the agent to make all its guesses for a target NMR spectra. All episodes are run within 300 seconds for 71.8% of the molecules and within 600 seconds for 88.5% of the molecules. It is observed in Figure 7 b. that the mean time taken for all the episodes for a molecule that is guessed correctly is ≈ 305 seconds, whereas the mean time for molecules that are guessed incorrectly is ≈ 780 seconds. This difference of distribution can be used to have more reliable predictions and improve the potential practical use-case of this framework. Stopping the search at a threshold time can improve accuracy for predicted molecules while also saving computational expense. When the framework makes predictions for all the molecules, i.e without any threshold time, the correct structure is among the guesses made for 94.8% of the molecules. Having a threshold time of 300 seconds leads to the framework making predictions for 72% of the molecules and timing-out for the rest of the molecules. The correct structure is among these guesses for 99% of the molecules. Similarly, when the threshold is set to 1000 seconds, the framework makes predictions for 94% of the molecules. The correct structure is among the guesses for 97% of the time.

Conclusion

This manuscript provides a framework using graph convolution networks and reinforcement learning to solve the inverse molecular problem of NMR spectra. We introduce a novel method to train the policy and value networks *a priori* in guided MCTS runs (Training Mode on) and demonstrate the utility of Monte Carlo Tree Searches in navigating the chemical space. Unlike other prior attempts to solve this problem like (Jonas 2019) where the model makes a prediction only 50% of the time, or (Zhang et al. 2020) where the model is tested only on 9 hand-picked target spectra, our model shows good promise by predicting the correct structure among its Top3 guesses, $\sim 80\%$ of the time.

Still, there are various avenues for improvement for future work. Since the RL algorithm is dependent on the forward model for its intermediate reward, a better scoring function would potentially improve the prediction accuracy. In the event that a better, albeit computationally expensive scoring function is devised, the overall practical accuracy can still be improved while being more time efficient. This can be done by scoring only the TopN guesses of the agent with the time intensive scoring function. As can be seen in Figure 6 b, target spectra’s correct molecular structure is present in the Top7 of the guesses $> 85\%$ of the time for all the runs with $nmcts > 200$.

^{13}C NMR is just one of the many spectroscopy techniques that are widely used. For example, ^1H NMR spectroscopy has a higher signal to noise ratio owing to the significantly larger abundance of spin-active isotope. Infrared spectroscopy sheds light on the vibrational transitions in a molecule and is considered to be complementary to NMR spectroscopy (Paulsen 2015) (Crowther 2008). A promising extension of work presented in this manuscript would be to incorporate other spectral data and leverage different kinds of information to elucidate the correct structure of an unknown molecule. Finally, we believe that this work is can be a crucial step in high-throughput synthesis, where swift and efficient verification of structures generated can make the whole process of drug discovery more robust and reliable.

References

- Asami, S.; Schmieder, P.; and Reif, B. 2010. High Resolution 1H-Detected Solid-State NMR Spectroscopy of Protein Aliphatic Resonances: Access to Tertiary Structure Information. *Journal of the American Chemical Society*, 132(43): 15133–15135.
- Balazs, A. Y. S.; Davies, N. L.; Longmire, D.; Packer, M. J.; and Chiarparin, E. 2021. Nuclear magnetic resonance free ligand conformations and atomic resolution dynamics. *Magnetic Resonance*, 2(1): 489–498.
- Benz, M.; Molla, M. R.; Böser, A.; Rosenfeld, A.; and Levkin, P. A. 2019. Marrying chemistry with biology by combining on-chip solution-based combinatorial synthesis and cellular screening. *Nature communications*, 10(1): 1–10.
- Bremser, W. 1978. Hose — a novel substructure code. *Analytica Chimica Acta*, 103(4): 355–365.
- Cherinka, B.; Andrews, B. H.; Sánchez-Gallego, J.; Brownstein, J.; Argudo-Fernández, M.; Blanton, M.; Bundy, K.; Jones, A.; Masters, K.; Law, D. R.; Rowlands, K.; Weijmans, A.-M.; Westfall, K.; and Yan, R. 2019. Marvin: A Tool Kit for Streamlined Access and Visualization of the SDSS-IV MaNGA Data Set. *The Astronomical Journal*, 158(2): 74.
- Crowther, M. W. 2008. NMR and IR Spectroscopy for the Structural Characterization of Edible Fats and Oils. An Instrumental Analysis Laboratory. *Journal of Chemical Education*, 85(11): 1550.
- Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; and Adams, R. P. 2015. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.
- Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. *CoRR*, abs/1704.01212.
- Jonas, E. 2019. Deep imitation learning for molecular inverse problems. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Jonas, E.; and Kuhn, S. 2019. Rapid prediction of NMR spectral properties with quantified uncertainty. *Journal of Cheminformatics*, 11(1): 50.
- Koichi, S.; Arisaka, M.; Koshino, H.; Aoki, A.; Iwata, S.; Uno, T.; and Satoh, H. 2014. Chemical structure elucidation from 13C NMR chemical shifts: Efficient data processing using bipartite matching and maximal clique algorithms. *Journal of chemical information and modeling*, 54(4): 1027–1035.
- Kuhn, S.; and Schlörer, N. E. 2015. Facilitating quality control for spectra assignments of small organic molecules: nmshiftdb2 – a free in-house NMR database with integrated LIMS for academic service laboratories. *Magnetic Resonance in Chemistry*, 53(8): 582–589.
- Landrum, G. 2016. RDKit: Open-Source Cheminformatics Software.
- Liu, R.; Li, X.; and Lam, K. S. 2017. Combinatorial chemistry in drug discovery. *Current Opinion in Chemical Biology*, 38: 117–126. Next Generation Therapeutics.
- Lodewyk, M. W.; Siebert, M. R.; and Tantillo, D. J. 2012. Computational prediction of 1H and 13C chemical shifts: a useful tool for natural product, mechanistic, and synthetic organic chemistry. *Chemical Reviews*, 112(3): 1839–1862.
- Mills, N. 2006. ChemDraw Ultra 10.0 Cambridge-Soft, 100 CambridgePark Drive, Cambridge, MA 02140. www.cambridgesoft.com. Commercial Price: \$1910 for download, \$2150 for CD-ROM; Academic Price: \$710 for download, \$800 for CD-ROM. *Journal of the American Chemical Society*, 128(41): 13649–13650.
- Moerland, T. M.; Broekens, J.; Plaat, A.; and Jonker, C. M. 2018. Monte Carlo Tree Search for Asymmetric Trees. arXiv:1805.09218.
- Paulsen, K. 2015. Yin and yang in chemistry education: the complementary nature of FTIR and NMR spectroscopies.
- Ramdas, A.; Garcia, N.; and Cuturi, M. 2015. On Wasserstein Two Sample Testing and Related Families of Nonparametric Tests. arXiv:1509.02237.
- Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T. P.; Simonyan, K.; and Hassabis, D. 2017. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *CoRR*, abs/1712.01815.
- Spronk, C. A.; Linge, J. P.; Hilbers, C. W.; and Vuister, G. W. 2002. Improving the quality of protein structures derived by NMR spectroscopy**. *Journal of Biomolecular NMR*, 22(3): 281–289.
- Vliegthart, J.; van Kuik, J.; and Hård, K. 1992. A 1H NMR database computer program for the analysis of the primary structure of complex carbohydrates. *Carbohydrate research*, 235: 53–68.
- You, J.; Liu, B.; Ying, Z.; Pande, V.; and Leskovec, J. 2018. Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Zhang, J.; Terayama, K.; Sumita, M.; Yoshizoe, K.; Ito, K.; Kikuchi, J.; and Tsuda, K. 2020. NMR-TS: de novo molecule identification from NMR spectra. *Science and Technology of Advanced Materials*, 21(1): 552–561. PMID: 32939179.
- Zhou, Z.; Kearnes, S. M.; Li, L.; Zare, R. N.; and Riley, P. 2018. Optimization of Molecules via Deep Reinforcement Learning. *CoRR*, abs/1810.08678.