Deep Reinforcement Learning for Molecular Inverse Problem of 2 Nuclear Magnetic Resonance Spectra to Molecular Structure

by

bhuvanesh sridharan, Sarvesh Mehta, Yashaswi Pathak, U Deva Priyakumar

in

Journal of Physical Chemistry Letters, 2022

Report No: IIIT/TR/2022/-1

Centre for Computational Natural Sciences and Bioinformatics
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2022
Deep Reinforcement Learning for Molecular Inverse Problem of Nuclear Magnetic Resonance Spectra to Molecular Structure

Bhuvanesh Sridharan, Sarvesh Mehta, Yashaswi Pathak, and U. Deva Priyakumar

ABSTRACT: Spectroscopy is the study of how matter interacts with electromagnetic radiation. The spectra of any molecule are highly information-rich, yet the inverse relation of spectra to the corresponding molecular structure is still an unsolved problem. Nuclear magnetic resonance (NMR) spectroscopy is one such critical technique in the scientists’ toolkit to characterize molecules. In this work, a novel machine learning framework is proposed that attempts to solve this inverse problem by navigating the chemical space to find the correct structure given an NMR spectra. The proposed framework uses a combination of online Monte Carlo tree search (MCTS) and a set of graph convolution networks to build a molecule iteratively. Our method can predict the structure of the molecule ∼80% of the time in its top 3 guesses for molecules with <10 heavy atoms. We believe that the proposed framework is a significant step in solving the inverse design problem of NMR spectra.

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 chemical characterization. It is an analytical technique based on the nuclei’s magnetic properties that have either an odd mass number or an even mass number with an odd atomic number. Nuclei with nonzero spin $\vec{S}$ would always have a nonzero magnetic dipole moment, $\vec{\mu}$. NMR relies on this for the nuclei to respond to electromagnetic waves as perturbations in the presence of an external magnetic field. In addition to small organic molecules, NMR spectroscopy is a critical method to obtain high-resolution information about proteins, DNA, and RNA. It can also be used to obtain knowledge of energy minima and barriers by observing conformational dynamics of proteins. This can be pivotal in the process of drug discovery.

The $^{13}$C NMR spectra measure the properties of individual nuclei and consist of peaks that correspond to each carbon 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 laboratories, experts manually identify the molecular structure from the NMR spectra using highly specific domain knowledge. To date, 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. These methods restrict the usage to identifying only those molecules that are stored in the database.

The problem under consideration here is a nonlinear inverse problem. The 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 $y$. One of the first attempts in the literature at recognizing and modeling this problem as an inverse problem was done by Jonas (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 to deep learning to solve the task. Other empirical methods, such as featuring the neighborhood of a nuclei and then matching it against a database of known motifs to predict its shift, are also common. 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.

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 a high-throughput manner. In this work, an effort has been made to determine the structure of a molecule given its NMR spectra and molecular

Received: March 1, 2022
Accepted: May 24, 2022
There have been a few endeavors to solve this inverse problem. Zhang et al.19 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 was able to predict the structure for six out of nine given spectra. In a work by Jonas,6 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. The work was tested on molecules with up to 32 heavy atoms.

In this work, we use a combination of online Monte Carlo tree search (MCTS)20 and a set of offline trained graph convolutional networks21 to navigate through the chemical space and find the correct molecular structure of a given target 13C NMR spectra.
To reiterate, the problem is defined in the following way: given $^{13}$C NMR spectra of a molecule consisting of each carbon’s shift and split values and the molecular formula, identify the structure of the molecule. The process of solving the problem is modeled as a Markov decision process (MDP) wherein the molecule is built iteratively from scratch by adding atoms and bonds to the current structure at each step. In this section, different components and details about the proposed framework are explained. The first part contains information about the data set used. The Letter gives details of the reinforcement learning algorithm used followed by information about the neural networks that aid the RL algorithm. Finally, the last paragraph of this section introduces a novel methodology of training prior and value networks.

We use nmrshiftdb2, which is a database for organic structures and their experimentally observed $^{13}$C NMR spectra. In this work, we consider only organic molecules that have less than 10 non-hydrogen atoms (C, O, N, and F). Charged molecules and radicals are also excluded. Thus, the data set comprises a total of 2134 molecules with experimentally obtained chemical shift and split values of $^{13}$C NMR spectra.

In this subsection, we attempt to define the state space and action space for the MDP comprehensively. This is followed by information about the agent which chooses the appropriate actions at a particular state. Choosing an accurate and appropriate measure of reward is essential for any RL algorithm to perform well. We use a forward NMR prediction model to formulate a reward function defined at the end of this subsection.

- **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 $n - s + 1$ components, one is a connected component representing the molecule of the current state, and the rest of the $n - s$ components are individual atoms that may join the current molecule by addition of new bonds later on. Here, a component is a subgraph which does not have any outgoing edges to the rest of the graph.

- **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 atomic number of the element that the node represents and the current valency of that atom that is available for further addition of bonds. A Gaussian, with the peak of the assigned shift centered at the chemical shift value and $\sigma = 2$, is discretized into 64 bins. This feature is then appended to the node feature.

- Because this work uses $^{13}$C NMR spectra as an input, it is certain that the target molecule contains at least one carbon atom. Hence, without the loss of generality, we choose to start building our molecule from a molecular

![Figure 4](image1.png)

Figure 4. (a) Target state of cyclohexane and current state of 2-methylbutane 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.

![Figure 5](image2.png)

Figure 5. An example run for the target molecule CC1=CC=NO1 with nmcts = 1000. $\pi_{tree}(a|s_j)$ 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 figure, each state $s_j$ is also accompanied by the splitvector of that state.
state containing a single carbon atom; that is, \( S_0 \) is just a carbon atom.

In this 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:

- At least one of the end points of the newly added edge must belong to the subgraph containing the 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. (Note 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 because this helps prune some redundant branches of the tree search.)

Because the problem is formulated 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 \( s \). A typical RL algorithm has two components: an agent, and an environment. The task assigned to the "agent" is to choose an action given the current state. On the other hand, the role of the "environment" is to simulate the action which was chosen by the agent and return the reward for the action which was taken. 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 node. The UCT value at any state is calculated based on the following formula:

   \[
   UCT(s, a) = Q(s, a) + c \times \frac{\pi_{\text{model}}(a|s)}{N(s)} \sqrt{\frac{\ln N(s)}{n(s, a)}} + 1
   \]

   where \( s \) is the current state, \( Q(s, a) \) the mean action value estimate, \( W(s, a) \) the cumulative of all returns \( R(s', a') \) until the leaf node, \( \pi_{\text{model}}(a|s) \) the prior probability by the policy network, \( N(s) \) the number of times state \( s \) has been visited.
reached, \( n(s, a) \) the number of times action \( a \) was taken from state \( s \), and \( c \) the constant with which one can manipulate the exploration versus exploitation ratio. The form of the UCT value used in this work is inspired by Moerland et al.,\textsuperscript{25} which was 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 \( s_L \). Because of computational limitations, this work uses a value neural network, \( V_{model}(s) \), 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.

\[
R(s_j, a_j) = R(s_j, a_j) + \gamma R(s_{j+1}, a_{j+1})
\]

The above four steps are repeated for \( n_{mcts} \) number of times. Then, a real action \( a_i \) 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 \( s_0 \).

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\textsuperscript{8} 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 the current state is to the target molecule. The forward model predicts the NMR spectra of the current state, and the reward is defined by

---

**Figure 7.** Target molecule and Top3 guesses as returned by the agent along with the reward and their predicted spectra for cases when the Top1 guess is the correct guess.
The reward \( r(s', a) = r(s') = 2 \left( \frac{1}{2} - \text{WS}(s') \right) \)

\[
\text{WS}(s') = \text{first Wasserstein distance} = l_1(S_T, S_C)
\]

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

where \( U \) and \( V \) are the CDFs for the distribution of some random variables \( u \) and \( v \), \( S_T \) is the NMR spectra of the target molecule, and \( S_C \) is the NMR spectra of the current molecules'.

The reward \( r(s') = \text{WS}(s') \) 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. Because 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.

Figure 8. Target molecule and Top3 guesses as returned by the agent along with the reward and their predicted spectra for cases when the Top1 guess is the incorrect guess.
Figure 9. (a) Accuracy when the model is trained on 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).

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

**Graph Featurizer:** This module uses a message passing neural network, 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 $e_{uv}$. 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:

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

$$h_v^{t+1} = U'_v(h_v^t, m_v^{t+1})$$

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

where $N(v)$ is the set of neighboring nodes of $v$, $M_v$ and $U'_v$ 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 $f_{model}(als)$ and value network $V_{model}(s)$.

**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 $V_{model}(s)$ of the current state.

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 data set for the prior policy neural network $f_{model}(als)$ and value network $V_{model}(s)$. When any action $a$ at state $s$ leads it to state $s'$, i.e., $s \xrightarrow{a} s'$, then

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

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

Each shift value in the data set 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 splittensor be the vector that stores the information about the number of carbons of each split kind in the current state. Because the only action possible in the modeled 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 splittensor would remain constant for states with only 1 connected component because the total number of carbons cannot 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 4a, the agent can safely terminate search through this branch because 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 4b, the agent can safely terminate because 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 prevent the tree from exploring branches that can lead to the incorrect structure.

The forward model used in this work was trained on nmrshiftdb2 data set22 as included in the original work by Jonas and Kuhn.25 The mean absolute error obtained for the prediction of the shift value per peak for the predictor was 1.374 ppm.

Figure 6 shows examples where the Top1 ranked molecule is the target molecule and there is no way to produce new quartet carbon atoms. These chemistry-guided conditions greatly prune the search tree and prevent the tree from exploring branches that can lead to the incorrect structure.

Even when the agent has the correct structure among its guesses, sometimes the scoring function ranks it lower than other guesses made, which reduces accuracy. Because the scoring function is dependent on the pretrained forward model, a more capable forward model is expected to increase the accuracy of the framework. In the event that a better, albeit computationally expensive scoring function is devised, the overall practical accuracy can still be improved while being even 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 6b, the target spectra’s correct molecular structure is present in the Top7 of the guesses >85% of the time for all the runs with nmcst > 200. In such a scenario, we can determine the correct structure for an NMR spectra by scoring only 7 structures while the MCTS search takes less than 100 s for most of the molecules.

In another experiment, molecules with <7 non-hydrogen atoms were filtered from the data set. 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 Figure 9a. However, the system performs well on the class of data that it was never exposed to before, guessing the correct structure 86.5% of the time.

The histogram of the time taken for the agent to run all the episodes for a molecule can be seen in Figure 6c. On average, it takes ~330 s for the agent to make all its guesses for a target NMR spectra. All episodes are run within 300 s for 71.8% of the molecules and within 600 s for 88.5% of the molecules. It is observed in Figure 9b that the mean time taken for all the 402 episodes for a molecule that is guessed correctly is ~305 s, whereas the mean time for molecules that are guessed incorrectly is ~780 s. 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 s 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 s, the framework makes predictions for 94% of the molecules. The correct structure is among the guesses 97% of the time.

This Letter provides a framework using graph convolution networks and reinforcement learning to solve the inverse molecular problem of NMR spectra. The work also introduces a novel method to train the policy and value networks a priori in guided MCTS runs (training mode on) and demonstrates the utility of Monte Carlo tree searches in navigating the chemical space. Unlike other prior attempts to solve this problem like the one by Jonas in which the model makes a prediction only 50% of the time (even though their work is tested on molecules with 32 heavy atoms), or the work by Zhang et al.28 in which the model is tested only on 9 hand-picked target spectra, this model shows good promise by predicting the correct structure among its Top3 guesses ~80% of the time. Additionally, it is observed that the proposed framework performs better than brute-force checking in an enumerated database of known molecular structures.

On average, it is seen that the framework calls the forward model
The authors thank IHub-Data, IIIT Hyderabad, and DST-SERB (CRG/2021/008036) for financial support.

REFERENCES


22. Kuhn, S.; Schlözer, N. E. Facilitating quality control for spectra assignments of small organic molecules: nrmsiftdb2 − a free in−

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpcl.2c00624.

AUTHOR INFORMATION

Corresponding Author

U. Deva Priyakumar − Centre for Computational Natural Science and Bioinformatics, International Institute of Information Technology, Hyderabad 500032, India; Email: deva@iiit.ac.in

Bhuvanesh Sridharan − Centre for Computational Natural Science and Bioinformatics, International Institute of Information Technology, Hyderabad 500032, India

Sarvesh Mehta − Centre for Computational Natural Science and Bioinformatics, International Institute of Information Technology, Hyderabad 500032, India

Yashaswi Pathak − Centre for Computational Natural Science and Bioinformatics, International Institute of Information Technology, Hyderabad 500032, India

Complete contact information is available at: https://pubs.acs.org/doi/10.1021/acs.jpcl.2c00624

Notes

The authors declare no competing financial interest.

The source code along with instructions for the work presented in this Letter can be found at https://github.com/devalab/SpectraToStructure.

ACKNOWLEDGMENTS

The authors thank IHub-Data, IIIT Hyderabad, and DST-SERB (CRG/2021/008036) for financial support.

https://doi.org/10.1021/acs.jpcl.2c00624

J. Phys. Chem. Lett. XXX, XXX, XXX−XXX
560 house NMR database with integrated LIMS for academic service
562 (23) Sridharan, B.; Goel, M.; Priyakumar, U. D. Modern Machine
563 Learning for Tackling Inverse Problems in Chemistry: Molecular
565 (24) Kaelbling, L. P.; Littman, M. L.; Moore, A. W. Reinforcement
569 (26) Ramdas, A.; Garcia, N.; Cuturi, M. On Wasserstein Two
570 Sample Testing and Related Families of Nonparametric Tests. *arXiv*
571 2015, 1509.02237.
573 Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks
576 (28) Landrum, G. *RDKit: Open-Source Cheminformatics Software;*
577 2016.