DeepSPInN - Deep reinforcement learning for molecular Structure Prediction from Infrared and ^{13}C NMR spectra

Sriram Devata,[†,§] Bhuvanesh Sridharan,[†,§] Sarvesh Mehta,[†,§] Yashaswi Pathak,[†] Siddhartha Laghuvarapu,[‡] Girish Varma,[¶] and Deva Priyakumar[*,†]

[†] *Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad, India*

[‡] *IHub-Data, International Institute of Information Technology, Hyderabad, India*

[¶] *Center for Security, Theory and Algorithms Research, International Institute of Information Technology, Hyderabad, India*

[§] *Contributed equally to this work*

E-mail: deva@iiit.ac.in;sriram.devata@research.iiit.ac.in

Abstract

Molecular spectroscopy studies the interaction of molecules with electromagnetic radiation, and interpreting the resultant spectra is invaluable for deducing the molecular structures. However, predicting the molecular structure from spectroscopic data is a strenuous task that requires highly specific domain knowledge. DeepSPInN is a deep reinforcement learning method that predicts the molecular structure when given Infrared and ^{13}C Nuclear magnetic resonance spectra by formulating the molecular structure prediction problem as a Markov decision process (MDP) and employs Monte-Carlo tree search to explore and choose the actions in the formulated MDP. On the

QM9 dataset, DeepSPInN is able to predict the correct molecular structure for 91.5% of the input spectra in an average time of 77 seconds for molecules with less than 10 heavy atoms. This study is the first of its kind that uses only infrared and ^{13}C nuclear magnetic resonance spectra for molecular structure prediction without referring to any pre-existing spectral databases or molecular fragment knowledge bases, and is a leap forward in automated molecular spectral analysis.

Introduction

Molecular spectroscopy is the analysis of the electronic, vibrational, and rotational excitations of the nuclei of molecules as they interact with electromagnetic radiation. It is widely used as a tool to identify and characterize molecules for quantitative and qualitative analysis of materials. The spectrum of a molecule is the measured absorption or emission of the incident electromagnetic radiation. Each molecule produces a unique spectrum for a particular spectroscopic method, allowing the spectrum to be used as a fingerprint of the molecule.

Infrared (IR) spectroscopy is a spectroscopic technique that sheds light on the vibrational modes of a molecule that changes its dipole moment.[1] These vibrational modes cause the molecules to absorb electromagnetic radiation in the Infrared spectral region, lying in the range of wavenumbers $4000 - 400$ cm^{-1}. Functional groups have unique absorbances in the region of peaks beyond 1500 cm^{-1} called the functional group region.[2] Peaks with wavenumbers < 1500 cm^{-1} are considered to be in the fingerprint region[2] since the elaborate patterns of peaks here are highly specific to a molecule and are often too complex to interpret.

Nuclear magnetic resonance (NMR) spectroscopy is another widely used spectroscopic technique to characterize the structure of molecules.[3] In NMR spectroscopy, an external magnetic field is applied to a molecule and the nuclei of some isotopes (e.g. ^{1}H, ^{13}C) absorb radio waves of specific frequencies to change their nuclear spin. In ^{13}C NMR for example, any small changes in the local environment of the atom in the molecule cause the ^{13}C nuclei to absorb radio waves of different frequencies. The relative differences of these frequencies against a reference ^{13}C NMR frequency of tetramethylsilane (TMS) are measured in parts per million (ppm)[4] to give the chemical shifts of the nuclei. The spin-spin coupling of the adjacent protons of the ^{13}C nuclei cause the splitting of the corresponding NMR signal and allows the calculation of the multiplicity of each peak. This chemical split of each ^{13}C nuclei's chemical shift is indicative of the number of directly attached hydrogen atoms. Together, the chemical shift and chemical split values of a ^{13}C NMR spectrum allow the deduction of the atom type and chemical environment of each carbon atom, and subsequently the complete structure of the molecule. The chemical split values however are difficult to obtain experimentally,[5] and are not used by DeepSPInN.

For a structure to be elucidated from molecular spectra, all structural fragments are identified by interpreting the peaks in the spectra as the first step. These structural fragments are combined to list the possible molecular structures that can be made. These structures are then verified by cross-referencing the expected peaks of the functional groups in the input spectra, or by comparing their predicted spectra with the input spectra. CASE (Computer Aided Structure Elucidation) programs have evolved a lot since their introduction and have made good progress for structure elucidation from spectra, but they are still expected to have a degree of intervention from chemists and spectrometrists.[6] These programs also typically require 2D spectra in addition to any 1D IR, NMR, and MS spectra as the input.[7] Even today, most computational methods to identify a substance from its spectral data rely on matching against a database of already known spectra or by searching through knowledge bases of substructures[8–16]. Such methods restrict their applicability to the cases where the molecule's spectra is already stored in the database, or cases where the structural motifs are adequately represented in the dataset. These database methods are also sensitive to variations in the experimental conditions while collecting the spectra,[14] and might fail if there are incorrect entries in the database.[17]

Recently, new methods have made use of Machine learning (ML) algorithms to solve problems in computational chemistry such as predicting new drug molecules[18–20], performing

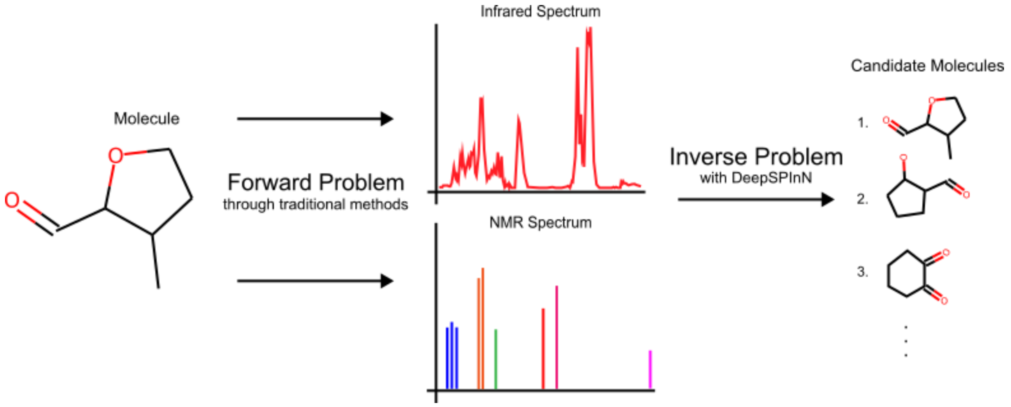


Figure 1: The IR and ^{13}C NMR spectra of 3-methyloxane-2-carbaldehyde to highlight the definitions of a *forward problem* and its corresponding *inverse problem*

molecular dynamics simulations[21–23], protein stability and binding site prediction[24,25], and predicting physical molecular properties.[26–28] Efforts for finding correlations between the spectral features of molecules and their structural features using ML can be dated back to the 1990s.[29] Interpretation of spectra to understand the complex relationship between a spectrum and the molecular structure is a difficult task. Recent developments in deep learning open new avenues to explore the mapping between the molecular structure and the information-rich spectral data.

The *forward problem* for molecular structure elucidation can be defined as the prediction of the spectra of a given molecular structure, and the corresponding *inverse problem* is generating the molecular structure given the spectra (Figure 1). Although they are computationally intensive, quantum mechanical methods can be used to obtain various molecular spectra. Many recent works made progress in solving the forward problem of predicting the spectra of a molecule where they utilize ML for predicting IR[30–34], NMR[35–37], UV-visible[38], and photoionization[39,40] spectra.

There have been works demonstrating how deep learning can solve inverse problems[41] in various domains. For the inverse problem in molecular structure elucidation, there have been works that aimed to automate the process of interpretation of IR spectra[42,43]. Many of them use only the functional group region of the spectra for their interpretation. Wang et al.[42] use a support vector machine to do multi-class classification for spectra from the OMNIC FTIR spectral library. The trained support vector machine identified 16 functional groups with a prediction accuracy of 93.3%. Fine et al.[43] introduce a multi-label neural network to identify functional groups present in a sample using a combination of FTIR and MS spectra. Jonas[44] and Howarth et al.[45] used a deep neural network that works with proton-coupled ^{13}C NMR to predict the molecular structure. Zhang et al.[46] use ChemTS[47] to identify a molecule from its NMR spectrum using Monte Carlo tree search (MCTS) guided by a recurrent neural network (RNN). Huang et al.[5] propose an ML-based algorithm that takes ^{1}H and ^{13}C NMR as input and predicts the correct molecule as the top scoring candidate molecule with an accuracy of 67.4%. Pesek et al.[48] introduce a rule based combinatorial approach in which the framework uses ^{1}H and ^{13}C NMR, IR, and mass spectra to elucidate the structure of an unknown compound and emphasises that the approach does not depend on database searches. Although this method does not use any spectral databases, it involves a step to pick ^{1}H NMR peaks and their multiplicities, which is subject to user interpretation and is heavily dependent on the correctness of the peak-picking step.[5] Such knowledge engineering and rule based approaches would limit the capability of the solution since they inherit the biases of the rules programmed,[14] and might not contain the data for fragments that are appropriate for the given input spectra.[49] This highlights the need for molecular structure elucidation methods that do not depend on spectral databases, while also not requiring any knowledge engineering.

Elyashberg and Argyropoulos[6] predict that using deep learning algorithms would improve the performance and robustness of CASE systems. They also highlight AlphaZero's success in mastering games[50] as a testament to how deep learning can learn to perform complicated tasks. A concurrent work[51] proposes a transformer model that utilizes IR spectra to achieve a top-1 accuracy of $\sim$ 55% on molecules with less than 10 heavy atoms. Another similar concurrent work[52] utilizes both ^{1}H and ^{13}C NMR spectra to achieve a top–1 accuracy of $\sim$ 70% on molecules with less than 10 heavy atoms. It has recently been shown that a

Monte-Carlo tree search (MCTS) algorithm can be used for the elucidation of molecular structure from ^{13}C NMR chemical shifts and splits, achieving a top-1 accuracy of 57.2%[53] for molecules with less than 10 heavy atoms on the nmrshiftdb2[54] dataset that contains experimentally calculated ^{13}C NMR spectra of 2134 molecules.

In this paper, our main contribution is a framework that utilizes IR and ^{13}C NMR spectra to accurately identify the molecular structure without any knowledge engineering or database searches. The proposed framework predicts the connectivity between the atoms, i.e. predicts the constitutional isomer of the molecular formula that corresponds to the input spectra. DeepSPInN formulates the molecular structure prediction problem as an MDP and employs MCTS to generate and traverse a search tree while using a set of pre-trained Graph Convolution Networks[55] to guide the tree search. DeepSPInN is able to achieve an accuracy of 91.5% on molecules with less than 10 heavy atoms, outperforming previous and concurrent works on structure elucidation from molecular spectra.

Methods

Dataset

The QM9[56,57] dataset is a subset of the GDB-17[58] chemical universe and consists of 134k stable small organic molecules with up to nine heavy atoms (CNOF). We first identified molecules in the QM9 dataset for which IR and ^{13}C NMR spectra were calculated using the Gaussian 09[59] suite of programs. We were able to calculate both IR and ^{13}C NMR spectra for 119,062 molecules. We then chose molecules where the smallest ring (if any ring(s) exist(s)) in the molecule has at least 5 atoms to account for ring strain, and molecules where none of the atoms have any formal charge. This left us with about 50k molecules to use as the input data for this work. A train-val-test split of 80-10-20 was used to make the train, validation, and test dataset of molecules. We used the validation set to choose hyperparameters for DeepSPInN, which we used for evaluating DeepSPInN on the test set.

To calculate the IR absorbance spectra, the geometrical optimization and the subsequent calculation of the vibrational frequencies were done using the B3LYP density functional methods with a 6-31g(2df,p) basis set in the gas phase. The spectra from these DFT calculations for each molecule is a set of frequency-intensity pairs. These infinitely sharp stick spectra were broadened to mimic actual gas-phase spectra using a peak broadening function as described and trained by McGill et al.[60]. This function is a two-layer fully connected neural network followed by an exponential transform, and takes frequency-intensity pairs to give a continuous spectrum. Following previous methods that predicted infrared spectra,[60] the intensities of the resulting spectra were binned with a bin-width of 2 cm^{-1} in the spectral range from $400 - 4000$ cm^{-1} to accommodate the available datasets of experimental infrared spectra. This results in the gas-phase IR absorbance spectrum for each molecule being represented by a 1801-length vector.

To analyse the congruence of the simulated and experimental IR spectra, we compare the simulated and experimental IR spectra of the molecules from our dataset that are also in the NIST Quantitative Infrared Database[61] and present this in the supplementary information. Due to the shortcomings of the DFT calculations and the peak expansion, the simulated spectra are not sufficiently similar to the experimental spectra to be considered as replacements for the experimental spectra. However, they reflect the complexity of experimental spectra by being able to account for the signatures of functional groups and by containing realistic peak shapes.[60,62] If DeepSPInN performs well by learning to capture relevant characteristics of simulated infrared spectra, it could similarly interpret and learn from experimental infrared spectra.

To make a dataset of ^{13}C NMR spectra, the peak positions (chemical shift) were obtained from the QM9-NMR dataset.[63] The QM9-NMR dataset has the gas phase mPW1PW91/6-311+G(2d,p)-level atom-wise isotropic shielding for the QM9 dataset. These ^{13}C isotropic shielding (σ_{iso}) values were converted to ^{13}C chemical shifts (δ_{iso}) through $\delta_{\text{iso}} = \sigma_{\text{iso}}^{\text{reference}} - \sigma_{\text{iso}}$,[64] where $\sigma_{\text{iso}}^{\text{reference}}$ is the reference value for tetramethylsilane (TMS), which is a standard reference compound. The root mean square error (RMSE) between the ^{13}C NMR spectra obtained in this way against spectra from the experimental nmrshiftdb2[54] database for the common molecules is 2.55 ppm per peak. As a reference, ^{13}C NMR shift values are typically between 0-200 ppm. The state-of-the-art ML-based ^{13}C NMR shift prediction methods achieve an RMSE of 1-5 ppm,[36,65,66] and DFT calculated ^{13}C NMR shift values have RMSE values ranging between 2.5–8.0 ppm.[67] An RMSE of 2.5 ppm shows great congruence between experimental ^{13}C NMR and the simulated ^{13}C NMR spectra that we use.

DeepSpInN Framework

The methods section is divided into five parts to explain the proposed framework:

i. description of how molecular structure prediction can be modelled as a Markov decision process (MDP)

ii. description of how MCTS can be used to generate a search tree of molecules and refine the policy at each state

iii. explanation of the architecture of the prior and value model used by DeepSPInN

iv. explanation of how ^{13}C NMR split values are used to prune the MCTS search tree

v. description of the training methodology used to train the prior and value model

MDP formulation

The problem of molecular structure prediction can be modelled as a finite Markov decision process (MDP)[68,69] in a way similar to the formulation in Sridharan et al.[53]. An MDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \{P_s\}, R \rangle$ with states $\mathcal{S}$, actions $\mathcal{A}$, policy $\{P_s\}$, and reward function R.[70] The goal is to learn the policies P_s which gives the transition probabilities over the action space $\mathcal{A}$ at a particular state $s \in \mathcal{S}$.

Each state $s \in \mathcal{S}$ consists of a molecular graph m and the target IR spectrum y_{IR}. A molecular graph represents a molecule where the atoms and bonds are mapped to nodes and edges in a graph. m also has the information about the target ^{13}C NMR spectrum encoded as node-wise features. In the initial state, the molecular graph is a null graph with nodes representing each atom in the molecular formula and no edges. The molecule mol_s at a state s is the largest connected component in the molecular graph. The remaining individual nodes in m might join mol_s after taking an action $a \in \mathcal{A}$. In the initial state, mol_s is just a single carbon atom corresponding to any of the nodes in m.

An action $a \in \mathcal{A}$ adds an edge between two nodes in m, which is equivalent to the addition of a bond between two atoms. Since the QM9 dataset has molecules that have a maximum of 9 atoms (number of nodes) and since there are 3 types of bonds (edges), the action space $\mathcal{A}$ has $9 * 9 * 3 = 243$ actions. For the molecular graphs to represent chemically valid molecules, only a subset of these actions can be considered to be valid. If a state has no valid actions that can be taken to reach any children states, it is a terminal state. In the action space for a state s, the valid actions are those that satisfy these conditions:

- Out of the two nodes that the action adds an edge between, at least one of the nodes must belong to the largest connected component (mol_s) of the molecular graph, i.e. the current molecule of the state.

- The edge added by the action should satisfy the chemical valency rules of the two nodes. If all the edges of a node do not satisfy the octet of the corresponding atom type, it is implicitly assumed that hydrogen atoms contribute to the octet.

- The action should not create a self-loop since atoms do not form bonds with themselves.

- The action does not add an edge between two nodes that already belong to the same cycle.

- The action does not create a cycle whose length is less than 5, since rings with less than 5 atoms have high ring strain if they have double or triple bonds.

The reward function $\mathcal{R}$ returns a non-zero reward for all terminal states and a zero reward for all non-terminal states. For the terminal states, the reward is a function of the spectral distance between the input IR spectrum and the IR spectrum of mol_s as predicted by Chemprop-IR.[60] Chemprop-IR is an extension to the Chemprop[71] architecture and uses a Directed Message Passing Neural Network[72](D-MPNN) to predict the IR spectrum of an input molecular graph. $\mathcal{R}$ is the Spectral Information Similarity[60] (SIS) metric which is calculated by rescaling the spectral divergence between two IR spectra found by their Spectral Information Divergence[73] (SID). The reward function $\mathcal{R}$ is given by:

$$\mathcal{R} = \text{SIS}(A, B) = \frac{1}{1 + \text{SID}(A, B)} = \left(1 + \sum_i (A_i \ln \frac{A_i}{B_i} + B_i \ln \frac{B_i}{A_i})\right)^{-1}$$

where A and B are two IR spectra.

Generating and exploring the search tree with MCTS

With this MDP formulation, we can use search algorithms to build a tree of state-labelled nodes.[74,75] We can build such a tree by repeatedly starting at the root state and reaching children states by taking any of the valid actions at each state. We use MCTS to estimate the optimal policy for the modelled reinforcement learning (RL) task.[76]

Starting from a root node, MCTS has 4 stages - selection, expansion, roll-out, and back-propagation (see Figure 2). In the selection stage, the algorithm chooses actions with probabilities proportional to their UCT[74] (Upper Confidence Bound applied to trees) values, until it reaches a leaf node. The UCT value of an action a at state s is given by

$$\text{UCT}(s, a) = Q(s, a) + c_{\text{puct}} \cdot \pi_s^a \cdot \frac{\sqrt{\sum_b N(s, b) + 1}}{N(s, a) + 1}$$

where $Q(s, a)$ is the expected reward of taking action a from state s, c_{puct} is a parameter to balance exploration and exploitation in the tree search, π_s^a is the probability of taking action a from state s according to the policy returned by a prior model, $N(s, a)$ is the number of

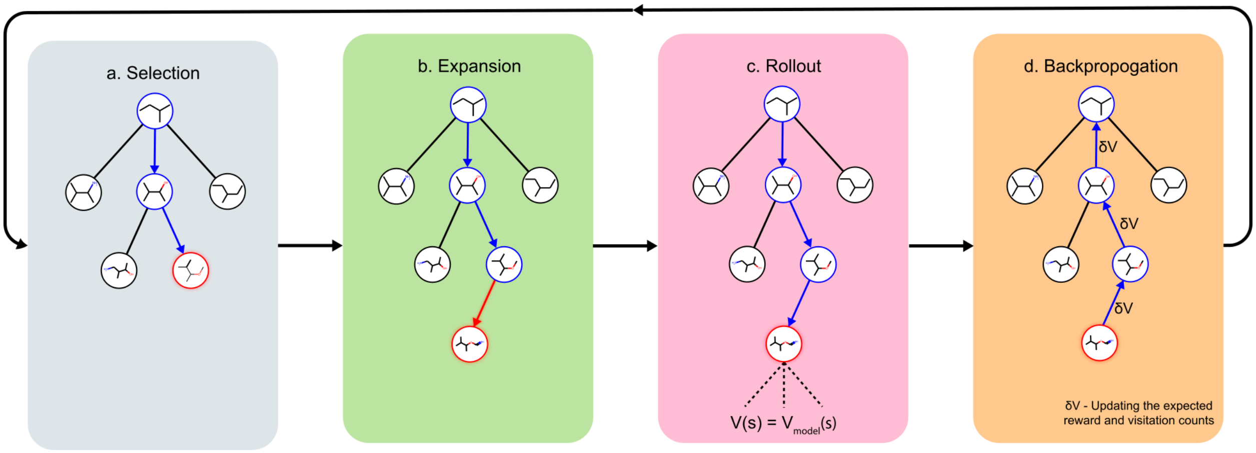


Figure 2: MCTS progresses in 4 stages to generate the search tree. a) Selection: starting from the root node of the tree, choose actions based on the UCT values b) Expansion: when the tree search reaches a leaf node, add a new child state to the tree c) Rollout: calculate the expected reward of the new child state through a series of random roll-outs d) Backpropagation: update the UCT values of all ancestors of the new child state

times action a has been taken from state s, and $\sum_b N(s,b)$ is the number of times state s has been reached.

In the process of traversing the search tree according to the UCT values, the algorithm would reach a point where taking an action a from state s would lead to a state s' that does not exist in the search tree. This leads to the expansion stage of MCTS where the new state s' is added to the search tree.

Once a new child node s' is added in the expansion stage, the rollout stage is used to evaluate the value of s'. An ideal way to calculate this value is to calculate the expected reward by a series of random rollouts. Due to the computational complexity of calculating the expected reward in the ideal way, we approximate the value using an offline-trained value model.[50,77] The value of s' is recursively back-propagated through all its parent nodes till the root node to update the ancestors' values and visitation counts. If s is a terminal state that already exists in the tree, the reward of s is back-propagated to update the values of all ancestor nodes. A state s is considered to be terminal if it has no valid actions, or if its reward exceeds a particular threshold (explained in the Supplementary Information). All 4 MCTS stages are repeated n_{mcts} number of times which is a hyper-parameter of DeepSPInN.

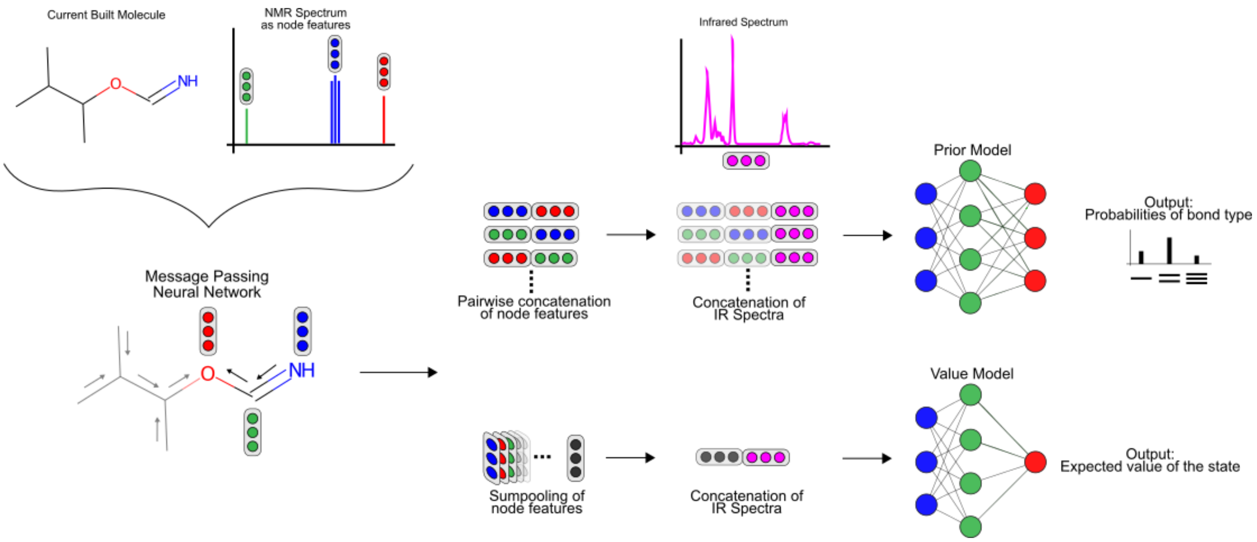

Current Built Molecule
NMR Spectrum as node features
NH
O
Infrared Spectrum
Message Passing Neural Network
Pairwise concatenation of node features
Concatenation of IR Spectra
Prior Model
Output: Probabilities of bond type
Value Model
Sumpooling of node features
Concatenation of IR Spectra
Output: Expected value of the state

Figure 3: A prior model and a value model are used with the MCTS algorithm to get the probabilities over the action space and to predict the value of a particular state. An MPNN uses the initial node-wise features that contain the ^{13}C NMR spectrum to give node-wise embeddings after three message passing steps. The prior model uses the pair-wise node embeddings and the IR spectrum to predict the probability of each pair of nodes having a single, double, or triple bond between them. The value model uses the sumpooled node-wise embeddings and the IR spectrum to predict the value of a particular state.

After n_{mcts} repetitions of the above 4 MCTS stages, a true action is taken according to the final policy at this state.

Description of the prior and value model

To featurize the built molecule at each state, both the prior and value model use a Message Passing Neural Network[55,78] (MPNN) that run for three time steps (see Figure 3). Consider a molecular graph $G(V, E)$ where each node has initial node features $x_v, \forall v \in V$. Each x_v is a vector of length 88 and contains the chemical description of the atom and the ^{13}C NMR peak of the atom corresponding to node v as listed in Table 1. Each node v also has hidden features h_v that are initialized to x_v, with the MPNN updating these hidden features in each time step of the forward pass. All edges in the molecular graph have edge features $e_{vw}, \forall v, w \in V$ as listed in Table 1. The forward pass of an MPNN has T message passing

[1] Used only for the experiment with proton-coupled ^{13}C NMR spectra

Table 1: Featurization of nodes and edges in the molecular graph

Node Feature	Description
Element Type	one-hot of [C,N,O,F]
Hybridization	one-hot of [sp, sp^2, sp^3]
Implicit Valency	one-hot of [0,1,2,3,4]
Radical Electrons	one-hot of [0,1,2]
Formal Charge	one-hot of [-2, -1, 0, 1, 2]
^{13}C NMR split	one-hot of [0,1,2,3] [1]
^{13}C NMR shift	a gaussian with $\sigma = 2$ centered at the chemical shift value discretized into 64 bins
Edge Feature	**Description**
Bond Type	one-hot of [single, double, triple, aromatic]
Bond Conjugation	boolean of whether the bond is conjugated
Presence in a Ring	boolean of whether the bond is in a ring

time steps and a final gathering step. The message passing steps use a message function M_t to form messages from the hidden features of neighbouring nodes $N(v)$ and the features of their corresponding edges. An update function U_t updates the hidden features of a node based on its current hidden features and the messages it received from its neighbouring nodes.

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

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

After T message passing steps, a gathering function G_T uses the initial node features x_v and the final hidden features h_v to give the node-wise features F_v.

$$F_v = G_T(x_v, h_v^t)$$

In DeepSPInN, M_t and U_t are fully connected neural neural networks, and G_T is an element-wise addition operation.

Using the node-wise features from the MPNN, the prior model generates all possible pairs of nodes and concatenates the node-wise features of all these pairs of nodes to get pair-wise features. y_{IR} is compressed into 100-length vectors by passing through a two-layer fully connected neural network to give y'_{IR} and is appended to all these pair-wise features. The product of this concatenation is passed through another two-layer fully connected neural network Pr_{model} to predict the probabilities of a bond of each of the three types (single, double, triple) existing between the pair of nodes. The prior model works as follows

$$P_{\mathrm{bond}} = Pr_{\mathrm{model}}\left([F_v, F_u, y'_{\mathrm{IR}}]\right), \text{ for each pair of nodes } u, v \in V$$

where, "[ ]" represents a concatenation operation, Pr_{model} is the prior model, and P_{bond} is a 3-tuple giving the probabilities of nodes u and v having a single, double, and triple bond respectively.

The value model first performs a sum-pooling operation on the node-wise features obtained from the MPNN. It then appends the compressed IR spectrum to the sum-pooled feature vector of the molecule and passes this through a two-layer fully connected neural network V_{model} to predict the value of this state. The value model works as follows

$$V_s = V_{\mathrm{model}}\left(\left[\sum_i F_i, y'_{\mathrm{IR}}\right]\right)$$

where, $\sum\limits_i F_i$ is the result of the sum-pooling operation of all node-wise features in the molecular graph.

Training Methodology

The prior and value model are trained on a set of experiences generated from a guided tree search on the molecules in the training dataset. These experiences are generated by building and exploring the search tree with MCTS, but with a modified reward function. Since the target molecule is known while training, the reward function is replaced with a binary function that returns a value depending on whether the molecule built at the current state is subgraph isomorphic to the molecular graph of the target molecule. The reward for taking an action a from state s to reach state s' is:

$$r(s,a) = \begin{cases} 1 & \text{if } S(\text{mol}_{s'}, \text{mol}_{target}) \\ 0 & \text{otherwise} \end{cases}$$

where $\text{mol}_{s'}$ is the molecular graph of the molecule at state s', mol_{target} is the molecular graph of the target molecule, and $S(\text{mol}_{s'}, \text{mol}_{target})$ is RDKit's[79] substructure search that does a subgraph isomorphism check and returns a boolean value.

The policies and values of each state in the trees built during the training period are stored and are used to train the prior and value models. We use the Adam optimizer[80] with a learning rate of $1e-5$ to train the models. The entire training took about 45 hours on a system with a Intel Xeon E5-2640 v4 processor and a GeForce GTX 1080 Ti GPU.

Choosing the hyperparameters n_{mcts} and number of episodes

We test multiple values of the n_{mcts} hyperparameter and the number of episodes for each set of input spectra to choose the best values. Each episode builds the MCTS tree from scratch by going through all four phases of MCTS n_{mcts} times and returns a final molecule. All the unique candidate molecules from these episodes are then ranked using the reward function as a scoring function. To choose the best hyperparameters, we consider the *Top N* metrics where each *Top N* metric denotes whether the target molecule was present in the top N ranked candidate molecules.

For the n_{mcts} hyperparameter, we test the values $200, 400$, and 800 on the validation set where each set of input spectra goes through a maximum of 40 episodes. The *Top N* metrics for each value of n_{mcts} is shown in table 2. Across the various n_{mcts} values, the *Top 1 (%)* accuracy increases as n_{mcts} increases. There is a stark increase in the *Top 1 (%)* accuracy between $n_{mcts} = 200, 400$, but there is only a marginal difference between

$n_{mcts} = 400, 800$. This shows that increasing n_{mcts} further will result in diminishing increase in performance while taking a disproportionately greater amount of time as shown in figure 5b. We use $n_{mcts} = 400$ to show the best results of DeepSPInN, and $n_{mcts} = 200$ to run various experiments in a reasonable time. To choose the number of episodes, we analyse the number of episodes that are taken when a molecule is correctly predicted. For the correctly predicted molecules, the right molecule was found within 10 episodes 86% of the time. The right molecule was found 99.9% of the time when DeepSPInN is run for 32 episodes, which we found to be the ideal number of episodes for running further experiments. Further information regarding this is provided in the supplementary information.

Table 2: *Top N* metrics for varying n_{mcts} values with 40 episodes on the validation set

	IR+^{13}C NMR		
n_{mcts}	200	400	800
Top 1 (%)	86.47	91.42	91.56
Top 3 (%)	87.05	92.13	92.49
Top 5 (%)	87.20	92.19	95.57
Top 10 (%)	87.39	92.33	96.07

Results

To rigorously evaluate DeepSPInN, we present the results of a few experiments in the following subsections. The first subsection compares the performance of DeepSPInN for different n_{mcts} values. The next subsection compares the final rewards for correctly and incorrectly predicted molecules. In the following subsection, the time taken to predict the molecules for different n_{mcts} values is analyzed. In the subsequent subsection, performance of the model is discussed when only one of IR or ^{13}C NMR spectrum is given as the input. The final subsection describes and presents the results of an experiment to check the generalizability of DeepSPInN.

Table 3: *Top N* metrics for varying n_{mcts} on the test set

	IR+^{13}C NMR	
n_{mcts}	200	400
Top 1 (%)	86.91	**91.46**
Top 3 (%)	87.54	**92.16**
Top 5 (%)	87.60	**92.22**
Top 10 (%)	87.62	**92.24**

Performance of DeepSPInN for varying n_{mcts} values

Table 3 compares the results for different values of n_{mcts} when given both IR and ^{13}C NMR spectra. For $n_{mcts} = 400$, DeepSPInN correctly identifies the target molecule $\sim$ 91.5% of the time as the top candidate molecule. Even with $n_{mcts} = 200$, DeepSPInN is able to outperform the previous MCTS-based structure elucidation method[53] that has a best *Top 1 (%)* accuracy of $\sim$ 60% compared to DeepSPInN's *Top 1 (%)* accuracy of $\sim$ 86.9% for $n_{mcts} = 200$.

Even within each n_{mcts} value, the *Top N (%)* metrics increase marginally starting from *Top 1 (%)* to *Top 10 (%)*. The increases across the *Top N (%)* metrics are due to an imperfect scoring function being used to rank all the candidate molecules. If the correct target molecule is not ranked as the top candidate molecule, it would contribute to one of the *Top N (%)* metrics. Still, we observe that the scoring function proposed in DeepSPInN is significantly better than the one used in Sridharan et al.[53] since they report great differences across the *Top N (%)* metrics. DeepSPInN does not show such great differences in the *Top N* metrics, illustrating that the scoring function used here performs better in ranking the candidate molecules. In DeepSPInN, if the correct molecule is found to be one of the candidate molecules, it is almost always ranked as the top candidate.

Comparison of rewards for correctly and incorrectly predicted molecules

Figure 4 contains the histograms of the rewards for the cases when DeepSPInN was and was not able to predict the correct molecule as the top candidate. The histogram of the rewards for the correctly predicted molecules has a very narrow distribution and has an average reward of 0.975. It is also left-skewed with most of the correctly predicted molecules receiving a higher reward when compared to the incorrectly predicted molecules. The histogram of the rewards for the incorrectly predicted molecules has a broader distribution with an average reward of 0.808. 88.56% of the correctly predicted molecules had a reward ≥ 0.95 while only 8.9% of the incorrectly predicted molecules had a reward ≥ 0.95. DeepSPInN would allow researchers to use the final reward as a confidence measure of the correctness of the prediction. When DeepSPInN gives a final reward ≥ 0.95 for a set of input spectra, the top candidate is the target molecule 99.9% of the time. The top candidate molecules even for these incorrectly predicted molecules are structurally similar to the correct molecule, with the average Tanimoto similarity between the correct molecule and the top candidate molecule being 0.954 for the test set.

Analysis of the time taken for the predictions

Figure 5a shows the distribution of times taken for DeepSPInN to predict candidate molecules for input IR and ^{13}C NMR spectra for different values of n_{mcts}. For $n_{\text{mcts}} = 400$, the average time taken is 77 seconds with 95% of the test molecules taking less than 130 seconds. Figure 5b shows the distributions of times taken by IR-and-NMR-trained, IR-trained, and NMR-trained models to predict candidate molecules for $n_{\text{mcts}} = 200$. The NMR-trained model has the fastest average prediction time of 24 seconds, while the IR-trained model has the slowest average prediction time of 82 seconds. The IR-and-NMR-trained model has an average prediction time of 49 seconds. The NMR-trained model is the fastest because the model is smaller due to the IR spectrum compression neural networks being removed. The IR-trained model is the slowest since DeepSPInN has to explore more of the search tree in each episode,

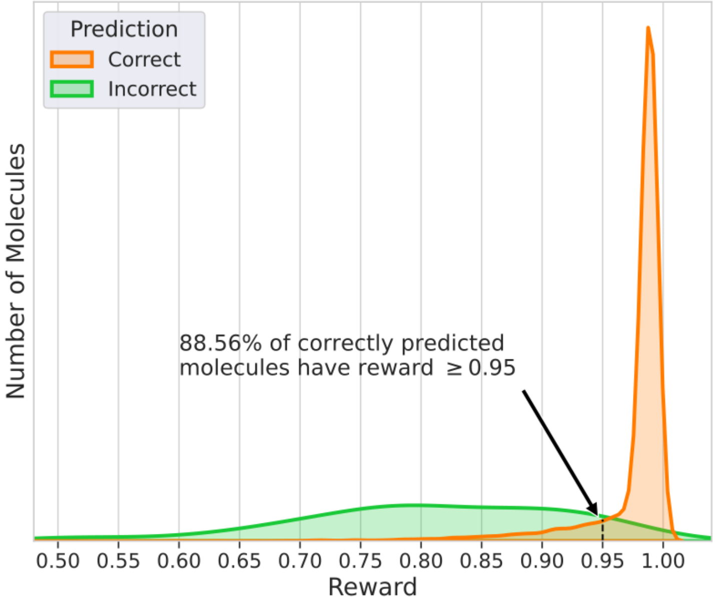
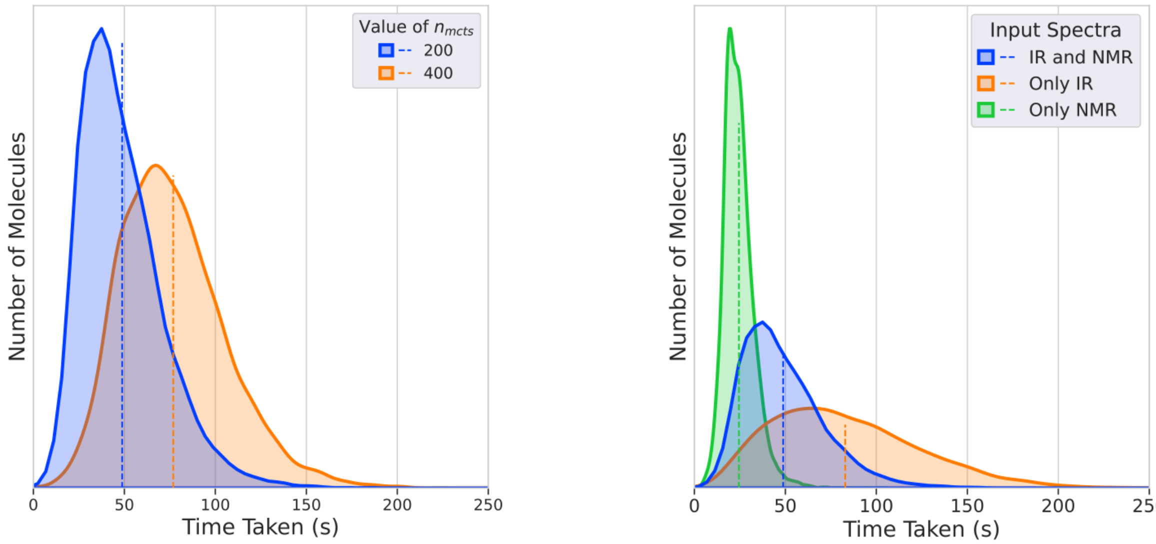


Figure 4: Histogram of the rewards of molecules that had the correct and incorrect structure as the top ranked candidate molecule for $n_{\text{mcts}} = 400$

when compared to the IR-and-NMR-trained model that also has the ^{13}C NMR shift values as the input.

Importance of having both IR and ^{13}C NMR spectra as input

To compare the distinguishing ability of IR and ^{13}C NMR and to compare the utility of having both IR and ^{13}C NMR spectra as the input, we performed ablation studies where we ran the model with either one of the spectra as the input for $n_{mcts} = 200$. Table 4 shows the *Top N* metrics for the models that received both IR and NMR, only IR, and only NMR spectra as input. The IR-and-NMR-trained model has a *Top 1* accuracy of 86.9% while the IR-trained and NMR-trained models have a *Top 1* accuracy of 73.15% and 29.37% respectively. All *Top N* metrics for the IR-and-NMR-trained model are greater than the models that work with either one of the spectra. This implies that the model is able to learn complementary information from both the spectra and subsequently performs better



(a) Histograms of the time taken to predict each molecule when given both IR and ^{13}C NMR spectra for varying n_{mcts} values

(b) Histograms of the time taken to predict each molecule when given either IR or ^{13}C NMR spectra for $n_{\text{mcts}} = 200$

Figure 5: Histograms of time taken to predict each molecule when given both IR and ^{13}C NMR spectra or either one spectrum

than the models with either one of the spectra as the input. Among the models that work on either one of the spectra, the IR-trained model performed significantly better than the NMR-trained model in all the *Top N* metrics.

Generalizability of DeepSPInN in understanding the action space

To understand how well DeepSPInN generalizes learning about the actions, the prior and value models were first trained on all molecules with less than 8 heavy atoms. It was then tested on a subset of molecules with 8 or more heavy atoms using these prior and value models. Table 5 shows the *Top N* metrics for this subset of test molecules, and the *Top N* metric for 8-atom molecules and 9-atom molecules in this subset. DeepSPInN achieves a *Top 1* accuracy of 68.52% even when all the test molecules have more heavy atoms than the molecules that DeepSPInN was trained on. The *Top 1* accuracy on molecules with 8 and 9 heavy atoms is 89.88% and 64.63% respectively. The decreased accuracy when compared to

Table 4: Performance of IR-and-NMR-trained, IR-trained, and NMR-trained models for $n_{mcts} = 200$

	IR and NMR	Only IR	Only NMR
Top 1 (%)	86.91	73.15	29.37
Top 3 (%)	87.54	73.31	37.99
Top 5 (%)	87.60	73.32	39.76
Top 10 (%)	87.62	73.32	40.66

Table 5: Training on molecules with ≤ 7 atoms and testing on molecules with ≥ 8 atoms for $n_{\text{mcts}} = 200$

	≥ 8 atom molecules	8-atom molecules	9-atom molecules
Top 1 (%)	68.52	89.88	64.63
Top 3 (%)	68.92	90.14	65.05
Top 5 (%)	69.0	90.27	65.12
Top 10 (%)	69.06	90.27	65.19

the original model might be because there were very few molecules for training the prior and value models in this experiment. When DeepSPInN is trained on molecules with ≤ 7 atoms, it might perform worse on bigger molecules since they have more combinations of functional groups in each test molecule than it has seen in the molecules used for the training. We study whether DeepSPInN is able to predict some functional groups better than the others by calculating the *Top N* for molecules that contain various functional groups. More details and results of both these experiments are available in the Supplementary Information.

Conclusions

DeepSPInN predicts the molecular structure when given an input IR and ^{13}C NMR spectra without searching any pre-existing spectral databases or enumerating the possible structural motifs present in the input spectra. After formulating the molecular structure prediction problem as an MDP, DeepSPInN employs MCTS to explore and choose the actions in the MDP. After building a null molecular graph from the molecular formula, DeepSPInN builds the molecular graph by treating the addition of each edge as an action in the MDP with the help of offline-trained GCNs to featurize each state in the MDP. DeepSPInN is able to correctly predict the molecular structure for 91.5% of input IR and ^{13}C NMR spectra in an average time of 77 seconds for molecules with < 10 heavy atoms.

DeepSPInN currently works on molecules that have less than 10 heavy atoms and future work could extend DeepSPInN to work on bigger molecules, or perhaps introduce other approaches that can easily be extended to bigger molecules. Since the number of molecules increases exponentially as the number of heavy atoms increase, future work could try to have a subset of molecules for different number of heavy atoms rather than trying to exhaustively train on all possible molecules of greater sizes. DeepSPInN currently requires the molecular formula to be inferred from another chemical characterization technique apart from the input spectra. Removing this requirement is an aspect that can be explored in the future. We demonstrated the capability of our method to effectively learn to characterize simulated IR and ^{13}C NMR spectra, which reflect the complexity of experimental spectra. This paves the way for future works to build datasets of experimental spectra and validate our method on them. Additionally, it will be interesting to see if DeepSPInN's accuracy improves with the addition of other spectral information such as UV-Vis spectra and mass spectra. We believe that DeepSPInN is a valuable demonstration of how machine learning can contribute to molecular structure prediction, and that it would help spur further research in the application of deep learning in high-throughput synthesis to enable faster and more efficient drug discovery pipelines.

Author contributions

S.D., B.S., and S.M. wrote the bulk of the code specific to this work. B.S., S.M., Y.P., and S.L conceptualized the original approach. S.D. prepared the draft manuscript text and figures with input from B.S., S.M, G.V., and U.D.P.. All co-authors were involved in editing and reviewing the final manuscript. U.D.P supervised the project.

Competing interests

The authors have been partly or entirely supported by IHub-Data, DST-SERB (CRG/2021/008036), and IIIT Hyderabad's Kohli Center on Intelligent Systems. The funders however did not have any role in the conceptualization, design, data collection, analysis, decision to publish, or preparation of the manuscript.

Correspondence

Any correspondence requests can be addressed to U.D.P. or S.D..

Data Availability

The data and code used in this work can be accessed through the GitHub repository.

References

(1) Atkins PW, d. P. J. *Elements of physical chemistry (5th ed.)*; Oxford: Oxford U.P, 2009; p 459.

(2) Janice Gorzynski Smith, S. J. *Organic Chemistry*; McGraw-Hill., Chapter 13 Mass Spectrometry and Infrared Spectroscopy.

(3) Brian E. Mann, B. F. T. *13C NMR data for organometallic compounds*; Academic Press, 1981.

(4) *The Theory of NMR - Chemical Shift*; University of Colorado, Boulder, Chemistry and Biochemistry Department, 2011.

(5) Huang, Z.; Chen, M. S.; Woroch, C. P.; Markland, T. E.; Kanan, M. W. A framework for automated structure elucidation from routine NMR spectra. *Chem. Sci.* **2021**, *12*, 15329–15338.

(6) Elyashberg, M.; Argyropoulos, D. Computer Assisted Structure Elucidation (CASE): Current and future perspectives. *Magnetic Resonance in Chemistry* **2021**, *59*, 669–690.

(7) Burns, D. C.; Mazzola, E. P.; Reynolds, W. F. The role of computer-assisted structure elucidation (CASE) programs in the structure elucidation of complex natural products. *Nat. Prod. Rep.* **2019**, *36*, 919–933.

(8) Steinbeck, C. Recent developments in automated structure elucidation of natural products. *Nat. Prod. Rep.* **2004**, *21*, 512–518.

(9) Kwon, Y.; Lee, D.; Choi, Y.-S.; Kang, S. Molecular search by NMR spectrum based on evaluation of matching between spectrum and molecule. *Scientific Reports* **2021**, *11*, 1–9.

(10) Dührkop, K.; Shen, H.; Meusel, M.; Rousu, J.; Böcker, S. Searching molecular structure databases with tandem mass spectra using CSI: FingerID. *Proceedings of the National Academy of Sciences* **2015**, *112*, 12580–12585.

(11) Elyashberg, M. E.; Williams, A.; Blinov, K. *Contemporary Computer-Assisted Approaches to Molecular Structure Elucidation*; New Developments in NMR; The Royal Society of Chemistry, 2012; pp P001–482.

(12) Hemmer, M. C.; Gasteiger, J. Prediction of three-dimensional molecular structures using information from infrared spectra. *Analytica Chimica Acta* **2000**, *420*, 145–154.

(13) Valli, M.; Russo, H. M.; Pilon, A. C.; Pinto, M. E. F.; Dias, N. B.; Freire, R. T.; Castro-Gamboa, I.; da Silva Bolzani, V. Computational methods for NMR and MS for

structure elucidation I: software for basic NMR. *Physical Sciences Reviews* **2019**, *4*, 20180108.

(14) Valli, M.; Russo, H. M.; Pilon, A. C.; Pinto, M. E. F.; Dias, N. B.; Freire, R. T.; Castro-Gamboa, I.; da Silva Bolzani, V. Computational methods for NMR and MS for structure elucidation II: database resources and advanced methods. *Physical Sciences Reviews* **2019**, *4*, 20180167.

(15) Bitchagno, G. T. M.; Tanemossu, S. A. F. Computational methods for NMR and MS for structure elucidation III: More advanced approaches. *Physical Sciences Reviews* **2019**, *4*, 20180109.

(16) Elyashberg, M.; Blinov, K.; Martirosian, E. A new approach to computer-aided molecular structure elucidation: the expert system Structure Elucidator. *Laboratory Automation & Information Management* **1999**, *34*, 15–30.

(17) Moreira, L. M. G.; Junker, J. Sampling CASE Application for the Quality Control of Published Natural Product Structures. *Molecules* **2021**, *26*, 7543.

(18) Vamathevan, J.; Clark, D.; Czodrowski, P.; Dunham, I.; Ferran, E.; Lee, G.; Li, B.; Madabhushi, A.; Shah, P.; Spitzer, M.; Zhao, S. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery* **2019**, *18*, 463–477.

(19) Ekins, S.; Puhl, A. C.; Zorn, K. M.; Lane, T. R.; Russo, D. P.; Klein, J. J.; Hickey, A. J.; Clark, A. M. Exploiting machine learning for end-to-end drug discovery and development. *Nature Materials* **2019**, *18*, 435–441.

(20) Mehta, S.; Laghuvarapu, S.; Pathak, Y.; Sethi, A.; Alvala, M.; Priyakumar, U. D. MEMES: Machine learning framework for Enhanced MolEcular Screening. *Chem. Sci.* **2021**, *12*, 11710–11721.

(21) Manzhos, S.; Carrington, T. Neural Network Potential Energy Surfaces for Small Molecules and Reactions. *Chemical Reviews* **2021**, *121*, 10187–10217.

(22) Pattnaik, P.; Raghunathan, S.; Kalluri, T.; Bhimalapuram, P.; Jawahar, C. V.; Priyakumar, U. D. Machine Learning for Accurate Force Calculations in Molecular Dynamics Simulations. *The Journal of Physical Chemistry A* **2020**, *124*, 6954–6967.

(23) Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C. Machine Learning for Molecular Simulation. *Annual Review of Physical Chemistry* **2020**, *71*, 361–390, PMID: 32092281.

(24) Samaga, Y. B. L.; Raghunathan, S.; Priyakumar, U. D. SCONES: Self-Consistent Neural Network for Protein Stability Prediction Upon Mutation. *The Journal of Physical Chemistry B* **2021**, *125*, 10657–10671.

(25) Aggarwal, R.; Gupta, A.; Chelur, V.; Jawahar, C. V.; Priyakumar, U. D. DeepPocket: Ligand Binding Site Detection and Segmentation using 3D Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **2021**,

(26) Pathak, Y.; Laghuvarapu, S.; Mehta, S.; Priyakumar, U. D. Chemically Interpretable Graph Interaction Network for Prediction of Pharmacokinetic Properties of Drug-Like Molecules. *Proceedings of the AAAI Conference on Artificial Intelligence* **2020**, *34*, 873–880.

(27) Laghuvarapu, S.; Pathak, Y.; Priyakumar, U. D. BAND NN: A Deep Learning Framework for Energy Prediction and Geometry Optimization of Organic Small Molecules. *Journal of Computational Chemistry* **2020**, *41*, 790–799.

(28) Goel, M.; Raghunathan, S.; Laghuvarapu, S.; Priyakumar, U. D. MoleGuLAR: Molecule Generation Using Reinforcement Learning with Alternating Rewards. *Journal of Chemical Information and Modeling* **2021**, *61*, 5815–5826.

(29) Ricard, D.; Cachet, C.; Cabrol-Bass, D.; Forrest, T. P. Neural network approach to structural feature recognition from infrared spectra. *Journal of Chemical Information and Computer Sciences* **1993**, *33*, 202–210.

(30) Ren, H.; Li, H.; Zhang, Q.; Liang, L.; Guo, W.; Huang, F.; Luo, Y.; Jiang, J. A machine learning vibrational spectroscopy protocol for spectrum prediction and spectrum-based structure recognition. *Fundamental Research* **2021**, *1*, 488–494.

(31) Yao, K.; Herr, J. E.; Toth, D.; Mckintyre, R.; Parkhill, J. The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics. *Chem. Sci.* **2018**, *9*, 2261–2269.

(32) Kananenka, A. A.; Yao, K.; Corcelli, S. A.; Skinner, J. L. Machine Learning for Vibrational Spectroscopic Maps. *Journal of Chemical Theory and Computation* **2019**, *15*, 6850–6858.

(33) Gastegger, M.; Behler, J.; Marquetand, P. Machine learning molecular dynamics for the simulation of infrared spectra. *Chem. Sci.* **2017**, *8*, 6924–6935.

(34) Schütt, K. T.; Kessel, P.; Gastegger, M.; Nicoli, K. A.; Tkatchenko, A.; Müller, K.-R. SchNetPack: A Deep Learning Toolbox For Atomistic Systems. *Journal of Chemical Theory and Computation* **2019**, *15*, 448–455.

(35) F.M., P.; A., H.; F., M.; S., D.; M., C.; L., E. Chemical shifts in molecular solids by machine learning. *Nature Communications* **2018**, *9*, All Open Access, Gold Open Access, Green Open Access.

(36) Jonas, E.; Kuhn, S. Rapid prediction of NMR spectral properties with quantified uncertainty. *Journal of Cheminformatics* **2019**, *11*, 50.

(37) Yang, Z.; Chakraborty, M.; White, A. D. Predicting chemical shifts with graph neural networks. *Chem. Sci.* **2021**, *12*, 10802–10809.

(38) Ye, S.; Hu, W.; Li, X.; Zhang, J.; Zhong, K.; Zhang, G.; Luo, Y.; Mukamel, S.; Jiang, J. A neural network protocol for electronic excitations of ¡i¿N¡/i¿-methylacetamide. *Proceedings of the National Academy of Sciences* **2019**, *116*, 11612–11617.

(39) Ghosh, K.; Stuke, A.; Todorović, M.; Jørgensen, P. B.; Schmidt, M. N.; Vehtari, A.; Rinke, P. Deep Learning Spectroscopy: Neural Networks for Molecular Excitation Spectra. *Advanced Science* **2019**, *6*, 1801367.

(40) S., K. G.; U., S.; J.M., R. Purifying Electron Spectra from Noisy Pulses with Machine Learning Using Synthetic Hamilton Matrices. *Physical Review Letters* **2020**, *124*, Cited by: 7; All Open Access, Green Open Access, Hybrid Gold Open Access.

(41) Ongie, G.; Jalal, A.; Metzler, C. A.; Baraniuk, R. G.; Dimakis, A. G.; Willett, R. Deep Learning Techniques for Inverse Problems in Imaging. 2020; `https://arxiv.org/abs/2005.06001`.

(42) Wang, Z.; Feng, X.; Liu, J.; Lu, M.; Li, M. Functional groups prediction from infrared spectra based on computer-assist approaches. *Microchemical Journal* **2020**, *159*, 105395.

(43) Fine, J. A.; Rajasekar, A. A.; Jethava, K. P.; Chopra, G. Spectral deep learning for prediction and prospective validation of functional groups. *Chem. Sci.* **2020**, *11*, 4618–4630.

(44) Jonas, E. Deep imitation learning for molecular inverse problems. Advances in Neural Information Processing Systems. 2019.

(45) Howarth, A.; Ermanis, K.; Goodman, J. M. DP4-AI automated NMR data analysis: straight from spectrometer to structure. *Chem. Sci.* **2020**, *11*, 4351–4359.

(46) Zhang, J.; Terayama, K.; Sumita, M.; Yoshizoe, K.; Ito, K.; Kikuchi, J.; Tsuda, K.

NMR-TS: de novo molecule identification from NMR spectra. *Science and Technology of Advanced Materials* **2020**, *21*, 552–561.

(47) Yang, X.; Zhang, J.; Yoshizoe, K.; Terayama, K.; Tsuda, K. ChemTS: an efficient python library for de novo molecular generation. *Science and technology of advanced materials* **2017**, *18*, 972–976.

(48) Pesek, M.; Juvan, A.; Jakos, J.; Kosmrlj, J.; Marolt, M.; Gazvoda, M. Database Independent Automated Structure Elucidation of Organic Molecules Based on IR, 1H NMR, 13C NMR, and MS Data. *Journal of chemical information and modeling* **2020**, *61*, 756–763.

(49) Elyashberg, M.; Blinov, K.; Molodtsov, S.; Smurnyy, Y.; Williams, A. J.; Churanova, T. Computer-assisted methods for molecular structure elucidation: realizing a spectroscopist's dream. *Journal of Cheminformatics* **2009**, *1*, 3.

(50) Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T.; Simonyan, K.; Hassabis, D. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. 2017; `https://arxiv.org/abs/1712.01815`.

(51) Alberts, M.; Laino, T.; Vaucher, A. C. Leveraging Infrared Spectroscopy for Automated Structure Elucidation. 2023.

(52) Alberts, M.; Zipoli, F.; Vaucher, A. C. Learning the Language of NMR: Structure Elucidation from NMR spectra using Transformer Models. 2023.

(53) Sridharan, B.; Mehta, S.; Pathak, Y.; Priyakumar, U. D. Deep Reinforcement Learning for Molecular Inverse Problem of Nuclear Magnetic Resonance Spectra to Molecular Structure. *The Journal of Physical Chemistry Letters* **2022**, 4924–4933.

(54) Kuhn, S.; Schlörer, N. E. Facilitating quality control for spectra assignments of small organic molecules: nmrshiftdb2 – a free in-house NMR database with integrated LIMS for academic service laboratories. *Magnetic Resonance in Chemistry* **2015**, *53*, 582–589.

(55) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. *CoRR* **2017**, *abs/1704.01212*.

(56) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling* **2012**, *52*, 2864–2875, PMID: 23088335.

(57) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data* **2014**, *1*, 140022.

(58) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *Journal of Chemical Information and Modeling* **2012**, *52*, 2864–2875.

(59) Frisch, M. J. et al. Gaussian 09, Revision A.1. 2016; Gaussian Inc. Wallingford CT.

(60) McGill, C.; Forsuelo, M.; Guan, Y.; Green, W. H. Predicting Infrared Spectra with Message Passing Neural Networks. *Journal of Chemical Information and Modeling* **2021**, *61*, 2594–2609.

(61) P.M., C.; F.R., G.; G.C., R.; W.J., L. The NIST Quantitative Infrared Database. *Journal of Research of the National Institute of Standards and Technology* **1999**, *104*, 59–81.

(62) Wallace, S.; Lambrakos, S. G.; Shabaev, A.; Massa, L. On using DFT to construct an IR spectrum database for PFAS molecules. *Structural Chemistry* **2022**, *33*, 247–256.

(63) Gupta, A.; Chakraborty, S.; Ramakrishnan, R. Revving up 13C NMR shielding predictions across chemical space: benchmarks for atoms-in-molecules kernel machine learning

with new data for 134 kilo molecules. *Machine Learning: Science and Technology* **2021**, *2*, 035010.

(64) Mehring, M. *High resolution NMR spectroscopy in solids*; Springer Science & Business Media, 2012; Vol. 11.

(65) Rull, H.; Fischer, M.; Kuhn, S. NMR shift prediction from small data quantities. *Journal of Cheminformatics* **2023**, *15*, 114.

(66) Guan, Y.; Shree Sowndarya, S. V.; Gallegos, L. C.; St. John, P. C.; Paton, R. S. Real-time prediction of 1H and 13C chemical shifts with DFT accuracy using a 3D graph neural network. *Chem. Sci.* **2021**, *12*, 12012–12026.

(67) Lodewyk, M. W.; Siebert, M. R.; Tantillo, D. J. Computational prediction of 1H and 13C chemical shifts: a useful tool for natural product, mechanistic, and synthetic organic chemistry. *Chem Rev* **2011**, *112*, 1839–1862.

(68) Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489.

(69) James, S.; Konidaris, G.; Rosman, B. An Analysis of Monte Carlo Tree Search. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence. 2017; p 3576–3582.

(70) Sutton, R. S.; Barto, A. G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press, 2018.

(71) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling* **2019**, *59*, 3370–3388.

(72) Dai, H.; Dai, B.; Song, L. Discriminative Embeddings of Latent Variable Models for Structured Data. *CoRR* **2016**, *abs/1603.05629*.

(73) Chang, C.-I. An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis. *IEEE Transactions on Information Theory* **2000**, *46*, 1927–1932.

(74) Kocsis, L.; Szepesvári, C. Bandit Based Monte-Carlo Planning. Machine Learning: ECML 2006. Berlin, Heidelberg, 2006; pp 282–293.

(75) Segler, M. H. S.; Preuss, M.; Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604–610.

(76) Moerland, T. M.; Broekens, J.; Plaat, A.; Jonker, C. M. Monte Carlo Tree Search for Asymmetric Trees. *arXiv preprint arXiv:1805.09218* **2018**,

(77) Silver, D. et al. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359.

(78) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. Advances in Neural Information Processing Systems. 2015.

(79) Landrum, G. RDKit: Open-Source Cheminformatics Software. **2016**,

(80) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. 2017.