Rethinking Structure Prediction in Computational Chemistry: The Role of Machine Learning in Replacing Database Searches

Thesis submitted in partial fulfillment
of the requirements for the degree of

*(Master of Science in **Computational Natural Sciences** by Research)*

by

Sriram Devata
2019113007
sriram.devata@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Rethinking Structure Prediction in Computational Chemistry: The Role of Machine Learning in Replacing Database Searches**" by **Sriram Devata**, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Prof. U Deva Priyakumar

To humanity's attempt to understand the universe and our place in it.

Acknowledgments

First and foremost, I extend my deepest gratitude to my parents, whose constant support and encouragement have been my cornerstone throughout my journey. Your sacrifices and faith in me have shaped me into the individual I am today, and this achievement is as much yours as it is mine.

A special thank you to Naren for being an extraordinary senior and a friend. Your friendship, guidance, and support have meant the world to me during these years. The experiences and learnings I've gained from our time together are treasures I'll always carry. A shout out to Arihanth for all the adventurous escapades; those moments of fun and adventure have been a delightful counterbalance to our rigorous academic and professional pursuits.

My heartfelt thanks go to Rahul and Ishaan as part of the RaIS group. The trips we have been on together and our Krunker and Valorant sessions were a source of great comfort and provided a much-needed sense of normalcy and belonging.

I am immensely grateful to the seniors in my lab - Siddhartha, Bhuvanesh, and Sarvesh - who have been pillars of support and guidance. Your mentorship has been invaluable, and I am thankful for the countless hours of advice, support, and collaboration we shared. I thank Prof. Deva Priyakumar for being my advisor.

I am grateful to everyone I've met and interacted with through the student bodies: Apex, The Dance Crew, E-Cell, Prithvi, Cult Council, Pentaprism, and Ping! . Each interaction has contributed to my growth and enriched my university experience in many ways.

Last but certainly not least, I thank Ahana for her patience and unwavering support. Your companionship and presence has been a source of joy, strength, and inspiration. Thank you for being there for me, celebrating my highs, and comforting me through my lows.

To all who have been a part of this journey, your contributions have not gone unnoticed. Thank you for being a part of my story.

Abstract

Well designed search algorithms can be used to search databases in computational chemistry to identify unknown compounds and their structures based on their observable attributes that are stored in the databases. Apart from an inherent problem of the lack of diversity within individual databases, algorithms that depend on database searches are inaccessible to researchers who are unable to access or have their own copy of these enormous databases. This thesis focuses on removing the database dependency for algorithms that depend on database searches for structure prediction in two areas within computational chemistry - molecular structure elucidation from molecular spectra, and tertiary structure prediction of RNA (Ribonucleic acid) molecules from their sequence.

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.

RNA molecules play a significant role in many biological pathways and have diverse functional roles, which is a result of their structural flexibility to fold into diverse conformations. This structural flexibility makes it challenging to obtain the structures of RNAs experimentally. Deep learning can be used to predict the secondary structures of RNA and other properties such as the backbone torsion angles, to be used as restraints for the computational optimization of the tertiary structures of RNA. TorRNA is a transformer encoder-decoder model, that takes an input RNA sequence and predicts the (pseudo)torsion angles of each nucleotide with a pre-trained RNA-FM model as the encoder. TorRNA is able to achieve a performance boost of $2\% - 16\%$ over the previous (pseudo)torsion angle prediction method for RNAs. We also demonstrate that TorRNA can used as a tool for model quality assessment of candidate RNA structures.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chemical structures of molecules and macromolecules play a critical role in determining the form and function of those molecules. The chemical structures can be thought of as Nature's blueprints to design molecules and macromolecules that take part in complex chemical and biological mechanisms [1]. Understanding these blueprints can help us design molecules with particular properties and interactions for applications like drug discovery, optoelectronics, energy storage, and designing better ways to synthesize compounds. A big step towards understanding these blueprints, is to gather data and learn about the chemical structures of currently useful molecules and their functionality.

In both experimental and computational chemistry, researchers have always sought to increase the amount of high-quality labelled data while ensuring its underlying diversity. Databases in chemistry contain data spanning a wide range of information ranging from materials and their properties to the results of computational simulations. Although these databases individually store a wealth of information, they all contain information that belongs to their own "niche" [2]. Users who require a diverse database would need to consolidate multiple individual databases, which proves to be a difficult task due to the high degree of variability of data formats [3] and heterogeneity of softwares used to obtain computational results [4]. Although there are efforts to encourage the usage of standard data formats, they are still works in progress and would require continued worldwide efforts. Databases should ideally accompanied with informative metadata and be built by following the findable, accessible, interoperable, recyclable (FAIR) principles, but building databases in such a way is not the norm yet [5].

These databases can serve as references to access first-principles results about the entries, or can be used in conjunction with well designed search algorithms to identify unknown compounds and their structures based on their observable attributes. Apart from the inherent problem of the lack of diversity within individual databases, algorithms that depend on database searches are inaccessible to researchers who are unable to access or have their own copy of these enormous databases. This thesis focuses on algorithms that depend on database searches for structure prediction in two areas within computational chemistry - molecular structure elucidation from molecular spectra, and tertiary structure prediction of RNA (Ribonucleic acid) molecules from their sequence.

Molecular spectroscopy is a frequently used analytical technique to identify the chemical structure of a compound. The spectroscopic data can be used as an 'address' or 'zip code' to locate entries in structure databases [6], resulting in a way to elucidate the structures based on molecular spectra. Current CASE (Computer Aided Structure Elucidation) programs provide good results for structure elucidation, but they are not automated and have heavy requirements on both the number of types of spectra and the pre-processing done on them [7]. Many of these CASE systems also rely on extensive databases of chemical structures and their molecular spectra. Even the largest spectroscopy databases currently in existence cover only a small percentage of all molecules with chemical or biological relevance. These databases do not represent the complexity or diversity of the vast chemical space, and can contain spectral data recorded in varying experimental settings. Methods that search through these databases can potentially not recognize unseen structural motifs [7] and would have trouble identifying unknown compounds, like truly novel drug metabolites [8].

To address this need for structure elucidation methods that do not depend on spectral databases and extensive preprocessing of the spectra, Chapter 2 of this thesis describes DeepSPInN - a Deep reinforcement learning method for molecular Structure Prediction from Infrared and ^{13}C NMR spectra [9]. On the QM9 dataset [10, 11], 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 < 10 heavy atoms.

Another area of computational chemistry where algorithms depend on database searches is for the tertiary structure prediction of macromolecules, with this thesis focusing on RNA molecules. Relatively recently, works like AlphaFold2 [12] have achieved near-experimental accuracy for protein structure prediction. This superior performance is attributed to the works' reliance on multiple sequence alignments (MSAs). MSAs provide co-evolution information of the input protein sequence, which is important for protein structure prediction [13]. MSAs can simply be seen as a list of protein sequences of the same length that are similar to the target protein chain sequence and share evolutionary information. These MSA-based algorithms see a decline in their performance when the quality of MSA degrades, and the step of searching for MSAs becomes a bottleneck in the structure prediction process. To remove this dependency on MSAs for protein structure predictions, some works [14] used a protein language model to learn the co-evolution information that is typically obtained from MSAs, or use representations of MSAs [15]. Recently, language models have been developed for RNAs [16] and have been used for end-to-end prediction of RNA structures [17], and demonstrate the potential of using these RNA language models for other RNA-related tasks.

To make further progress towards addressing the need of methods that predict the tertiary structure of RNA without performing computationally-heavy MSA, Chapter 3 of this thesis describes TorRNA - a method for improved prediction of Torsion angles of RNA. TorRNA is a transformer encoder-decoder model, that takes an input RNA sequence and predicts the (pseudo)torsion angles of each nucleotide with a pre-trained RNA language model as the transformer encoder. TorRNA is able to achieve a performance boost of $2\% - 16\%$ over the previous (pseudo)torsion angle prediction method and consequently shows an improved performance over a random baseline predictor as well.

Chapter 2

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

2.1 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 [18]. 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 [19]. Peaks with wavenumbers < 1500 cm^{-1} are considered to be in the fingerprint region [19] 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 [20]. In NMR spectroscopy, an external magnetic field is applied to a molecule and the nuclei of some isotopes (e.g. ^1H, ^{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) [21] 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 [22], 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 [23]. These programs also typically require 2D spectra in addition to any 1D IR, NMR, and MS spectra as the input [24]. 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 [25, 26, 27, 28, 29, 30, 31, 32, 33]. 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 [31], and might fail if there are incorrect entries in the database [34].

Recently, new methods have made use of Machine learning (ML) algorithms to solve problems in computational chemistry such as predicting new drug molecules [35, 36, 37], performing molecular dynamics simulations [38, 39, 40], protein stability and binding site prediction [41, 42], and predicting physical molecular properties [43, 44, 45]. Efforts for finding correlations between the spectral features of molecules and their structural features using ML can be dated back to the 1990s [46]. 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 2.1). 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 [47, 48, 49, 50, 51], NMR [52, 53, 54], UV-visible [55], and photoionization [56, 57] spectra.

There have been works demonstrating how deep learning can solve inverse problems [58] 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 [59, 60]. Many of them use only the functional group region of the spectra for their interpretation. Wang et al. [59] use a support vector machine to do multi-class classification for spectra from the OMNIC FTIR spectral library. The trained support vector

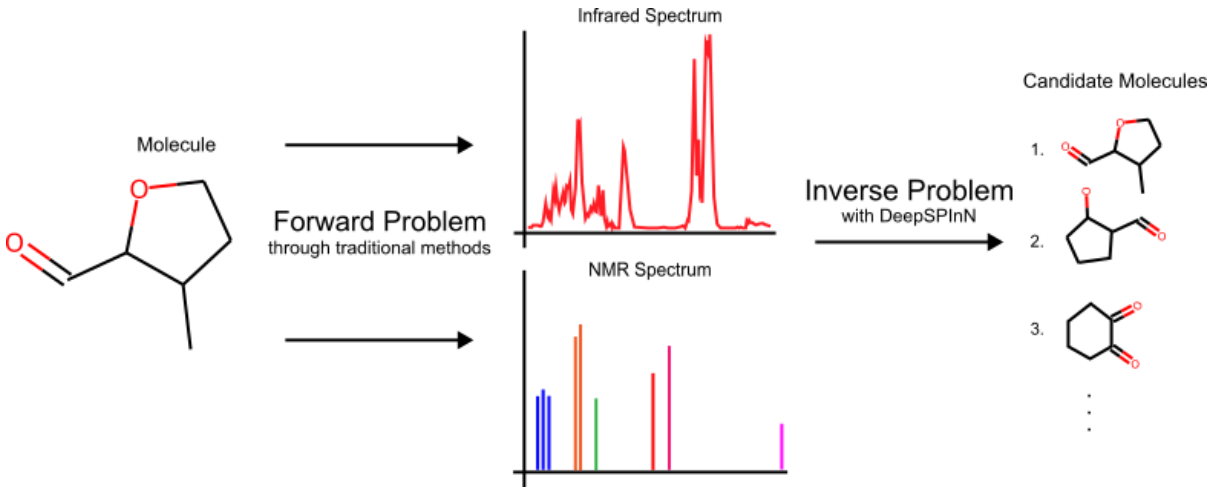


Figure 2.1: The IR and ^{13}C NMR spectra of 3-methyloxane-2-carbaldehyde to highlight the definitions of a *forward problem* and its corresponding *inverse problem*

machine identified 16 functional groups with a prediction accuracy of 93.3%. Fine et al. [60] introduce a multi-label neural network to identify functional groups present in a sample using a combination of FTIR and MS spectra. Jonas [61] and Howarth et al. [62] used a deep neural network that works with proton-coupled ^{13}C NMR to predict the molecular structure. Zhang et al. [63] use ChemTS [64] to identify a molecule from its NMR spectrum using Monte Carlo tree search (MCTS) guided by a recurrent neural network (RNN). Huang et al. [22] 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. [7] 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 [22]. Such knowledge engineering and rule based approaches would limit the capability of the solution since they inherit the biases of the rules programmed [31], and might not contain the data for fragments that are appropriate for the given input spectra [65]. 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 [23] predict that using deep learning algorithms would improve the performance and robustness of CASE systems. They also highlight AlphaZero's success in mastering games [66] as a testament to how deep learning can learn to perform complicated tasks. A concurrent work [67] 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 [68] 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% [69] for molecules with less than 10 heavy atoms on the nmrshiftdb2 [70] dataset that contains experimentally calculated ^{13}C NMR spectra of 2134 molecules.

In this thesis chapter, the 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 [71] 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.

2.2 Methods

2.2.1 Dataset

The QM9 [10, 11] dataset is a subset of the GDB-17 [72] 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 [73] 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 thesis. 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. [74]. 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 [74], 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 [75] and present this in the appendix. 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 [76, 74]. 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 [77]. 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 $\delta_{\text{iso}} = \sigma_{\text{iso}}^{\text{reference}} - \sigma_{\text{iso}}$ [78], 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 [70] 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 [53, 79, 80], and DFT calculated ^{13}C NMR shift values have RMSE values ranging between 2.5–8.0 ppm [81]. An RMSE of 2.5 ppm shows great congruence between experimental ^{13}C NMR and the simulated ^{13}C NMR spectra that we use.

2.2.2 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

2.2.2.1 MDP formulation

The problem of molecular structure prediction can be modelled as a finite Markov decision process (MDP) [82, 83] in a way similar to the formulation in Sridharan et al. [69]. 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 [84]. 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 [74]. Chemprop-IR is an extension to the Chemprop [85] architecture and uses a Directed Message Passing Neural Network [86](D-MPNN) to predict the IR spectrum of an input molecular graph. $\mathcal{R}$ is the Spectral Information Similarity [74] (SIS) metric which is calculated by rescaling the spectral divergence between two IR spectra found by their Spectral Information Divergence [87] (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.

2.2.2.2 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 [88, 89]. 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 [90].

Starting from a root node, MCTS has 4 stages - selection, expansion, roll-out, and back-propagation (see Figure 2.2). In the selection stage, the algorithm chooses actions with probabilities proportional to their UCT [88] (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

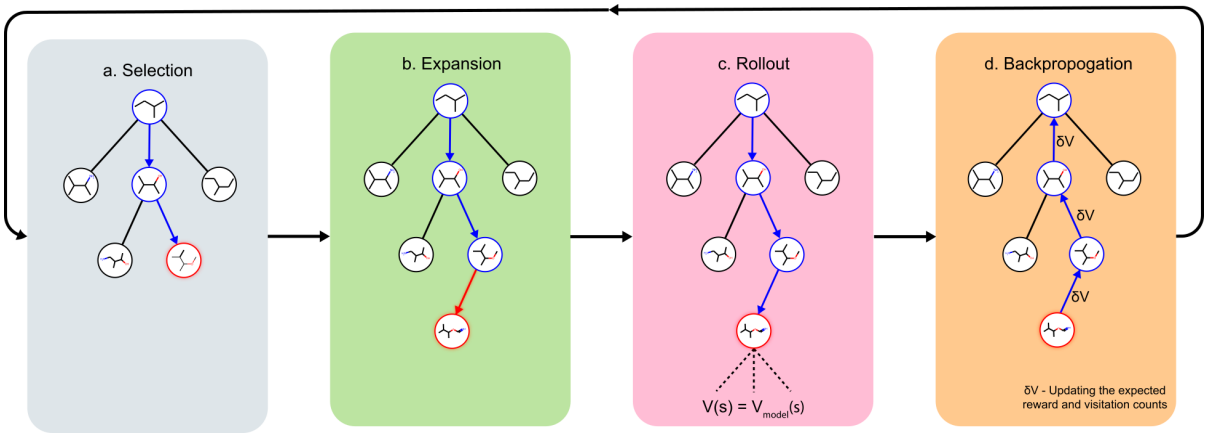


Figure 2.2: MCTS progresses in 4 stages to generate the search tree. a) Selection: starting from the root node of the tree, choose actions based on the UCT values b) Expansion: when the tree search reaches a leaf node, add a new child state to the tree c) Rollout: calculate the expected reward of the new child state through a series of random roll-outs d) Backpropagation: update the UCT values of all ancestors of the new child state

according to the policy returned by a prior model, $N(s, a)$ is the number of 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 [66, 91]. 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 appendix). All 4 MCTS stages are repeated n_{mcts} number of times which is a hyper-parameter of DeepSPInN. After n_{mcts} repetitions of the above 4 MCTS stages, a true action is taken according to the final policy at this state.

2.2.2.3 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 [71, 92] (MPNN) that run for three time steps (see Figure 2.3). Consider a molecular

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

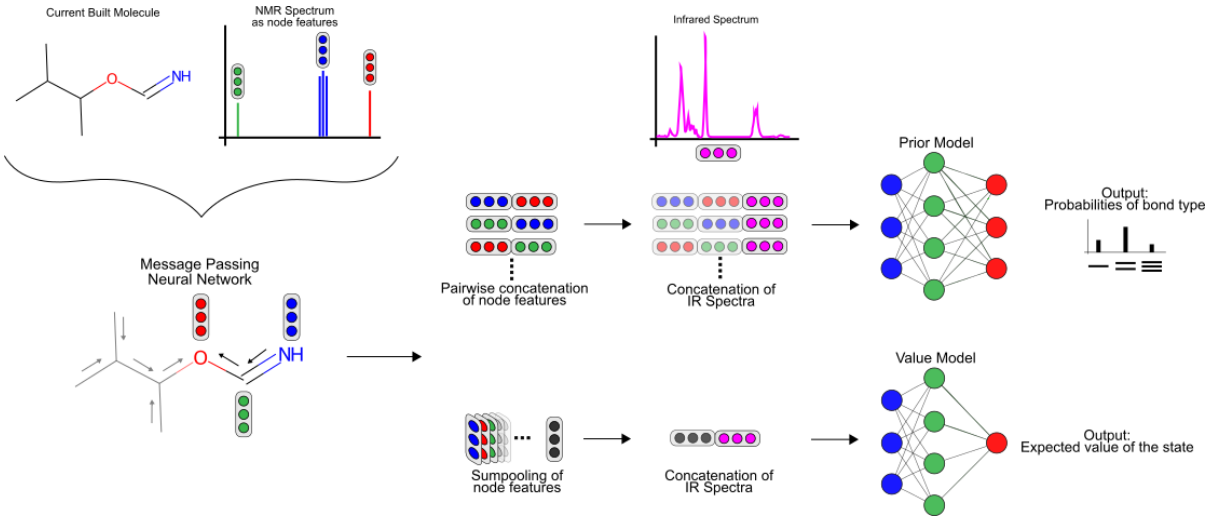


Figure 2.3: A prior model and a value model are used with the MCTS algorithm to get the probabilities over the action space and to predict the value of a particular state. An MPNN uses the initial node-wise features that contain the ^{13}C NMR spectrum to give node-wise embeddings after three message 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.

Table 2.1: 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

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 2.1. 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 2.1. The forward pass of an MPNN has T message passing 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

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

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

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

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

2.2.2.4 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:

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

where $\text{mol}_{s'}$ is the molecular graph of the molecule at state s', $\text{mol}_{\textit{target}}$ is the molecular graph of the target molecule, and $S(\text{mol}_{s'}, \text{mol}_{\textit{target}})$ is RDKit's [93] 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 [94] 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.

2.2.2.5 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 2.2. 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 2.5b. 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 appendix.

Table 2.2: *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

2.3 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

Table 2.3: *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**

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.

2.3.1 Performance of DeepSPInN for varying n_{mcts} values

Table 2.3 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 [69] 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. [69] 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.

2.3.2 Comparison of rewards for correctly and incorrectly predicted molecules

Figure 2.4 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

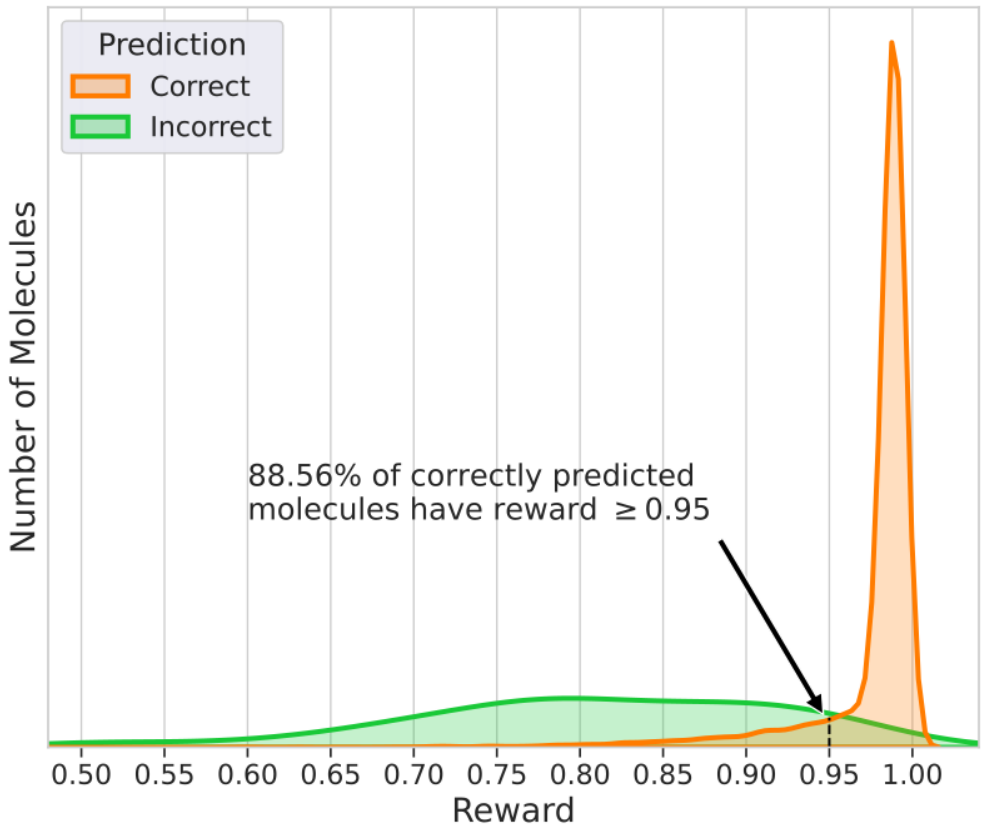
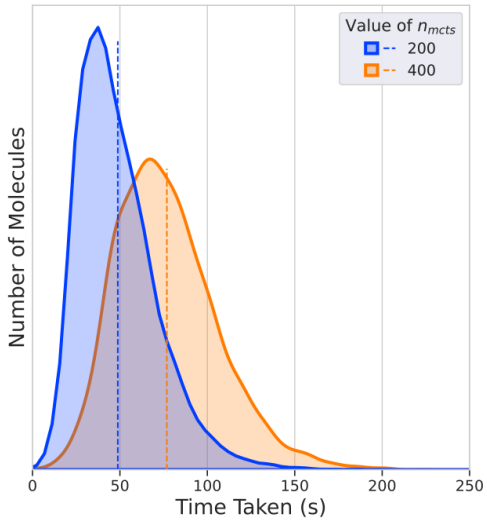
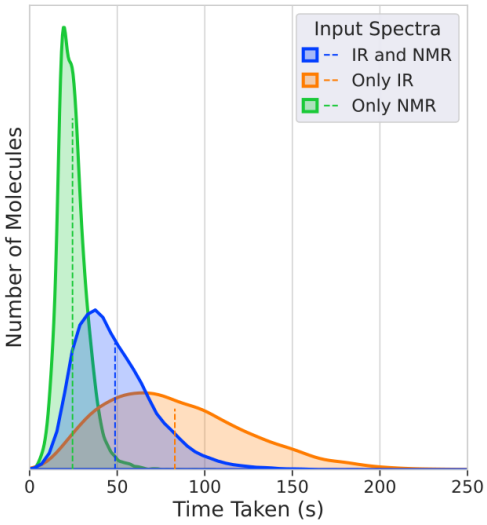


Figure 2.4: Histogram of the rewards of molecules that had the correct and incorrect structure as the top ranked candidate molecule for $n_{\text{mcts}} = 400$

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.



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

Table 2.4: 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

2.3.3 Analysis of the time taken for the predictions

Figure 2.5a 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 2.5b 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, when compared to the IR-and-NMR-trained model that also has the ^{13}C NMR shift values as the input.

2.3.4 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 2.4 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 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.

Table 2.5: 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

2.3.5 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 2.5 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 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 appendix. In another experiment shown in the appendix, the current DeepSPInN model trained on simulated spectra does not perform well on elucidating structures from experimental spectra. DeepSPInN is able to learn the complexity of spectra, as seen by its performance on simulated spectra, and can be generalized to perform well on unseen experimental spectra when it is also trained on experimental spectra.

2.3.6 Structural complexity of molecules resolved by DeepSPInN

To demonstrate the structural complexity addressed by DeepSPInN in elucidating molecular structures from Infrared and ^{13}C NMR spectra, we show 20 complex molecules that were the top candidate molecule as predicted by DeepSPInN in Figure 2.6. We quantified the complexity of molecules using the Bertz Complexity [95] descriptor implemented in RDKit [93].

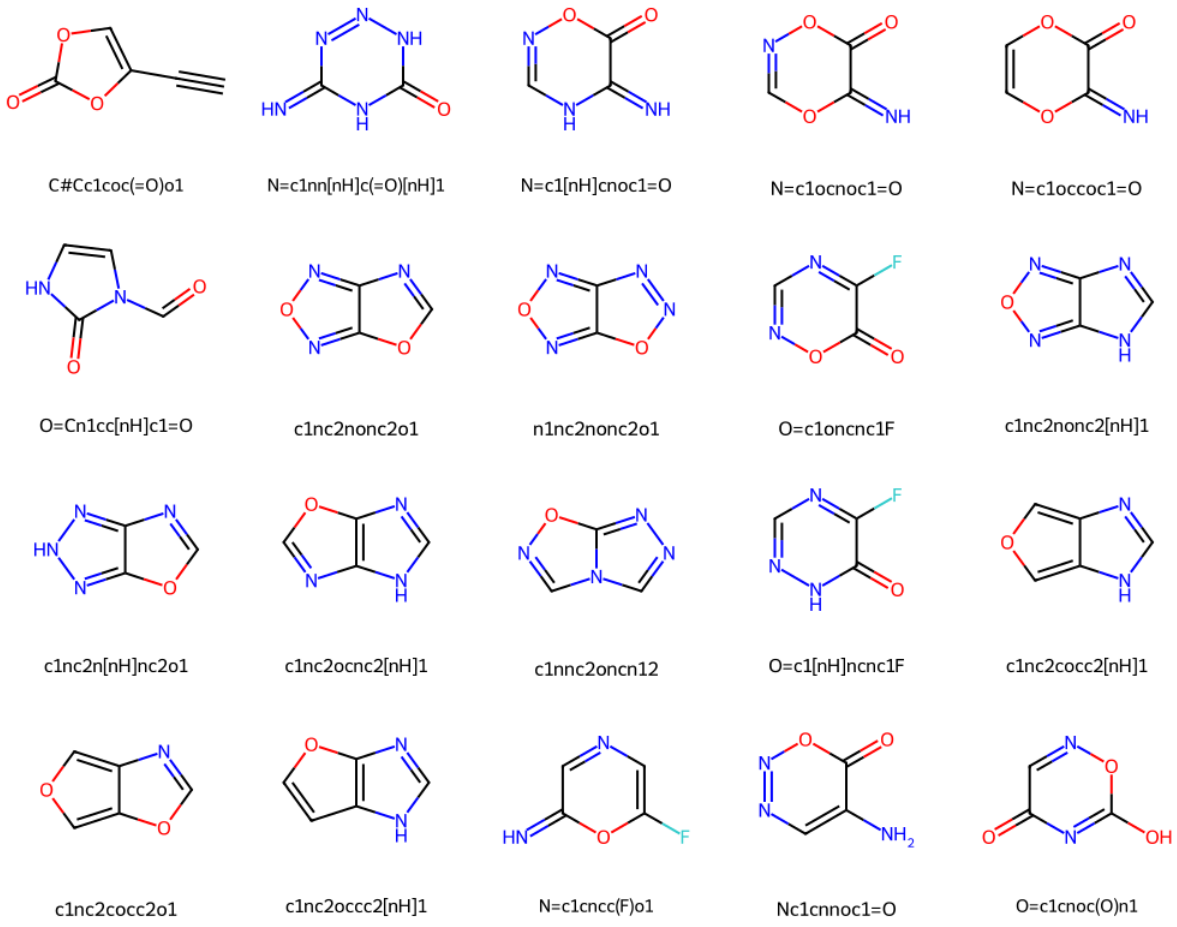


Figure 2.6: 20 complex molecules successfully predicted by DeepSPInN, demonstrating the structural complexity addressed by DeepSPInN

Chapter 3

TorRNA - improved prediction of Torsion angles of RNA by leveraging large language models

3.1 Introduction

RNA molecules play a significant role in modulating many biological pathways, ranging from acting as catalytic ribozymes [96] to controlling gene expression via transcriptional regulation [97]. Recent advances in generation [98] and delivery [99] of RNA make it more feasible for RNA molecules to be used as therapeutic agents [100] to address the underlying pathology of diseases rather than treating the symptomology as done by small molecule-based therapeutics [101]. RNA that are involved in disease pathways can also serve as druggable targets for small molecules to bind to the RNA and modulate their function [102], increasing the number of ways we can interfere with pathological mechanisms. This functional diversity of RNA molecules is closely tied to their structure, with their ability to fold into various conformations impacting how they interact with other molecules [103, 104, 105]. Determining the structures of RNA is important for understanding their mechanisms and to be able to exploit them as therapeutic agents and targets.

RNA molecules fold hierarchically with their secondary structure elements being folded first, which then interact and result in the tertiary structure [106]. RNA molecules fold into their secondary structures and specific sub-structures based on hydrogen bonding between the nucleotides and their stacking, to form helices and unique RNA loops like hairpin loops and pseudoknots. These secondary structure elements interact and form the tertiary structure, and result in the great structural plasticity exhibited by RNA molecules. Determining the tertiary structures of these RNA molecules through experimental means such as nuclear magnetic resonance and X-ray crystallography is challenging due to the resolution limits of these methods and the intrinsic structural plasticity of RNA molecules [107, 108].

To alleviate the struggles of determining the structure of RNA molecules experimentally, a number of computational approaches based on thermodynamic models and Watson-Crick-Franklin (WCF) interactions have been developed to determine the secondary structure of RNA molecules over the years [109, 110, 111, 112, 113, 114]. Recently, new methods have made use of Machine learning (ML) al-

gorithms to solve problems in computational chemistry such as predicting and synthesizing new drug molecules [35, 36, 37, 9], performing molecular dynamics simulations [38, 39, 40], protein stability and binding site prediction [41, 42], and predicting physical molecular properties [43, 44, 45]. ML has been employed to predict the secondary structure of RNA as early as the 1990s [115, 116, 117].

Recent advances in deep learning have resulted in improved prediction of macromolecular structures like proteins [12, 118] and RNA [119, 120, 121]. The breakthroughs in protein structure prediction by deep learning are due to the improved prediction of contact maps and backbone structures, which are used as restraints for modelling the structures. However, there are only a few studies that predict such restraints for RNA molecules [119, 122]. With existing methods optimizing the tertiary structure of RNA molecules when given the secondary structures, deep learning can be used to solve the downgraded problem of predicting the secondary structures and other structural properties [16] that can be used as restraints for the optimization. Presented in this thesis, TorRNA focuses on accurate prediction of the backbone structure of RNA molecules by predicting the torsion and pseudotorsion angles that can characterize the backbone of an RNA molecule.

In proteins, the backbone configuration can be described by only two backbone conformational parameters ϕ and ψ. For nucleic acid structures like RNA and DNA however, the phosphodiester backbone is best characterized by 6 torsion angles ($\alpha, \beta, \gamma, \delta, \epsilon,$ and ζ), and a torsion angle χ that quantifies the orientation of the base with respect to the sugar. For a nucleotide indexed i and the next nucleotide along the $5'-3'$ direction indexed as $i+1$, these 7 torsion angles as shown in Figure 3.1 can be described as the dihedral angle between the atoms $O3'_{i-1} - P_i - O5'_i - C5'_i(\alpha)$, $P_i - O5'_i - C5'_i - C4'_i(\beta)$, $O5'_i - C5'_i - C4'_i - C3'_i(\gamma)$, $C5'_i - C4'_i - C3'_i - O3'_i(\delta)$, $C4'_i - C3'_i - O3'_i - P_{i+1}(\epsilon)$, $C3'_i - O3'_i - P_{i+1} - O5'_{i+1}(\zeta)$, and $O4'_i - C1'_i - (N9_i/N1_i) - (C2_i/C4_i)(\chi)$. To simplify the representation of the RNA backbone configuration, two pseudotorsion angles eta (η) and theta (θ) can be used to describe the RNA backbone configuration [103, 123] similar to how ϕ and ψ are used to describe backbone configuration of proteins. These pseudotorsion angles as shown in Figure 3.1 can be described as the dihedral angle between the atoms $C4'_{i-1} - P_i - C4'_i - P_{i+1}(\eta)$ and $P_i - C4'_i - P_{i+1} - C4'_{i+1}(\theta)$ where $i-1$, i, and $i+1$ are the indices of three nucleotides in the $5' - 3'$ direction. These 9 torsion and pseudotorsion angles are depicted in Figure 3.1 and are henceforth referred to as (pseudo)torsion in the rest of the manuscript.

SPOT-RNA-1D [122] employed a residual dilated convolutional neural network architecture [124, 125] to predict seven torsion and two pseudotorsion angles, and was able to beat a random baseline predictor by achieving a mean absolute error (MAE) between 14° and 44° for the nine (pseudo)torsion angles. The design choice of using a dilated convolutional neural network architecture is justified by the architecture's ability to learn long-range interactions between nucleotides. However, SPOT-RNA-1D and other methods that predict secondary structures of RNA employ variations of CNNs since the properties they predict are represented by two-dimensional matrices - such as contact maps.

When compared to proteins, PDB [126] has fewer 3D RNA structures. This lack of RNA sequence-structure datapoints is one of the greatest challenge in developing ML-based sequence to structure methods for RNA. The RNA foundation model (RNA-FM) [16] is a foundation model trained in a

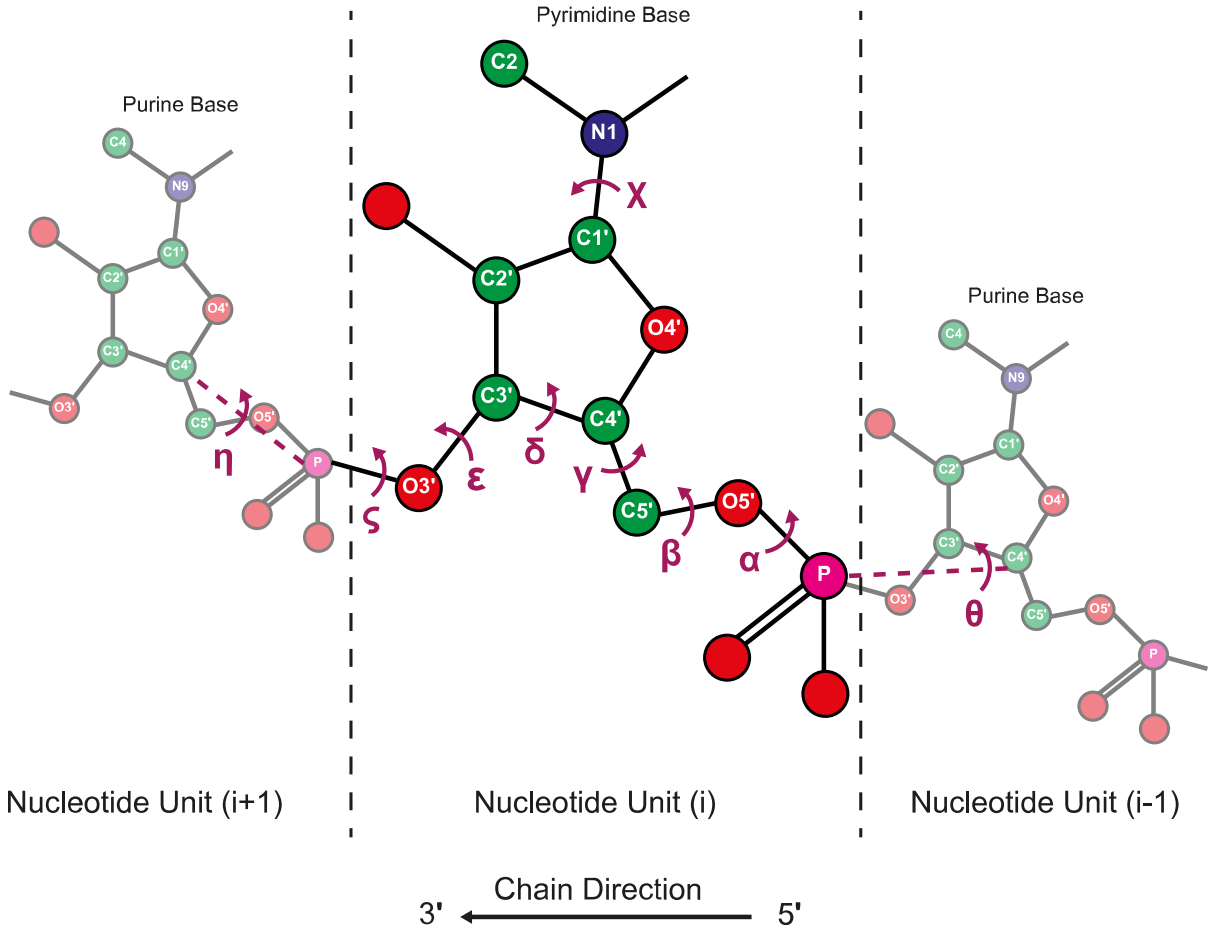


Figure 3.1: RNA backbone torsion $(\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \chi)$ and pseudotorsion (η, θ) angles.

self-supervised manner to learn any patterns in the RNA sequences and generates sequence encodings that potentially capture the underlying evolutionary, structural, and functional information of the corresponding RNA molecules from their sequences. RNA-FM implicitly learns the co-evolutionary information of RNA sequences from 23 million unlabeled non-coding RNA sequences and performed well in downstream tasks like RNA secondary structure prediction and 3D contact map prediction. RNA-FM has been used by E2Efold-3D [17] to develop the first end-to-end deep learning approach to predict 3D RNA structures directly from the sequence, highlighting the importance of the information contained in the RNA-FM encodings.

In this thesis, we present TorRNA - a method that focuses on predicting the (pseudo)torsion angles of each residue by using a transformer [127] architecture to predict the (pseudo)torsion angles. TorRNA uses the encodings of all nucleotides of an input RNA sequence as generated by a pre-trained RNA-FM model and predicts the (pseudo)torsion angles using a transformer decoder architecture. The choice of using a transformer is consistent with the choice of using a dilated convolutional neural network since a

transformer also contains residual connections [128] to help learn the long-range interactions between nucleotides.

3.2 Methods

3.2.1 Dataset

SPOT-RNA-1D's [122] training dataset contains 286 RNA chains, with the validation and test dataset containing 30 and 147 RNA chains respectively. However, this dataset was constructed by downloading all RNA structures from PDB [126] with a suitable X-ray resolution on October 3, 2020. To train and test TorRNA, we sought to create a new dataset that contains the RNA structures uploaded to PDB [126] in the recent years.

The dataset of RNAs used for training and testing TorRNA was curated with data from RCSB Protein Data Bank (PDB) [126] and BGSU RNA Representative Sets [129]. More specifically, we assembled the PDB identifiers of RNA structures that were available with a resolution of $< 4\mathring{A}$ from PDB on July 4, 2023 and from Release 3.288 of BGSU RNA Representative Sets. The structures of these RNAs were downloaded from PDB [126] using their PDB identifiers. The downloaded PDB structures are processed using the Biopython [130] package to obtain the structures of individual RNA chains.

We follow the same methodology as SPOT-RNA [119] to make the train, validation, and test splits of the dataset. To remove the redundancies in the dataset, the sequences of all the RNA chains with < 500 nucleotides were clustered using CD-HIT-EST [131] with a sequence identity threshold of 80%. The RNA sequences that do not belong to any clusters are assigned to a noncluster set (NCS), and the clustered RNA sequences are assigned to a cluster set (CS). To ensure an even stronger nonredundancy between NCS and CS, we run the BLAST-N [132] tool on the RNA sequences with an e-value cutoff of 10. Sequences in CS that have hits with sequences in NCS are removed to ensure that sequence homologies between CS and NCS are minimal. The resulting CS is used as the training data, and NCS is randomly divided into validation and test dataset with a 20-80 split.

While dividing NCS into the validation and test datasets, we maintained the RNA sequences from the RNA-Puzzles benchmarking test set [133, 134, 135, 136, 137] exclusively in the test dataset for TorRNA to run further experiments on these RNAs as described in the 3.3. The final training, validation, and test datasets have 767, 42, and 172 RNAs respectively. When comparing the performance of TorRNA with SPOT-RNA-1D [122] in the Results section, we use the same dataset splits used by SPOT-RNA-1D [122] in one of the results. For the list of curated PDB IDs, we use the DSSR [138, 139] software tool to calculate the native torsion angles and to identify the structural regions from the 3D structures. The final dataset is available on our code repository.

3.2.2 Architecture of TorRNA

TorRNA's overall architecture is a transformer encoder-decoder as shown in Figure 3.2a, that takes an input RNA sequence and predicts the (pseudo)torsion angles of each nucleotide. TorRNA utilizes a pre-trained RNA-FM [16] model's embedding layer and subsequent transformer encoder blocks [127, 140] to obtain encodings for each residue of an RNA sequence. RNA-FM's [16] model architecture as shown in Figure 3.2b is a stack of 12 transformer encoder blocks, similar to the BERT [140] language model architecture. Each encoder block has a hidden size of 640 and 20 self-attention heads, with layer normalization and residual connections being applied before and after every block. For an RNA sequence as the input, RNA-FM first tokenizes the sequence into the individual nucleotide tokens ('A', 'U', 'G', and 'C' among others). An initial embedding layer maps each of these sequential nucleotide tokens to 640-dimensional vectors. These initial embeddings are passed through the stack of 12 encoder blocks to give final encodings of the same size for each nucleotide. These final encodings of each nucleotide contain information aggregated from the entire RNA sequence.

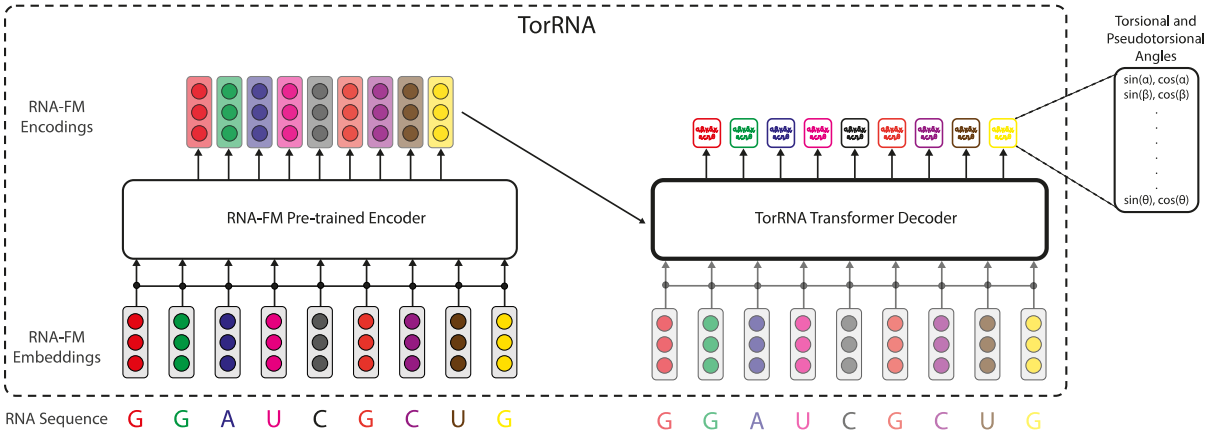
The final encodings of each nucleotide computed by the pre-trained RNA-FM model are then passed to a stack of 3 transformer decoder blocks [127] along with the embeddings of the nucleotides computed by the pre-trained embedding layer of RNA-FM. These decoder blocks as shown in Figure 3.2b use the embeddings of each nucleotide and perform cross-attention over the RNA-FM encodings to finally predict the (pseudo)torsion angles for each nucleotide. Since these angles are in the range $[-180°, 180°]$, TorRNA predicts the *sine* and *cosine* values of the (pseudo)torsion angles instead of predicting the angles directly to handle the periodicity of the angles as done in previous works that predict torsion angles for RNA and proteins [122, 141]. The predicted *sine* and *cosine* values can be used to calculate the angle using the inverse tangent function.

$$\textit{angle} = \tan^{-1}\left(\frac{\sin(\textit{angle})}{\cos(\textit{angle})}\right)$$

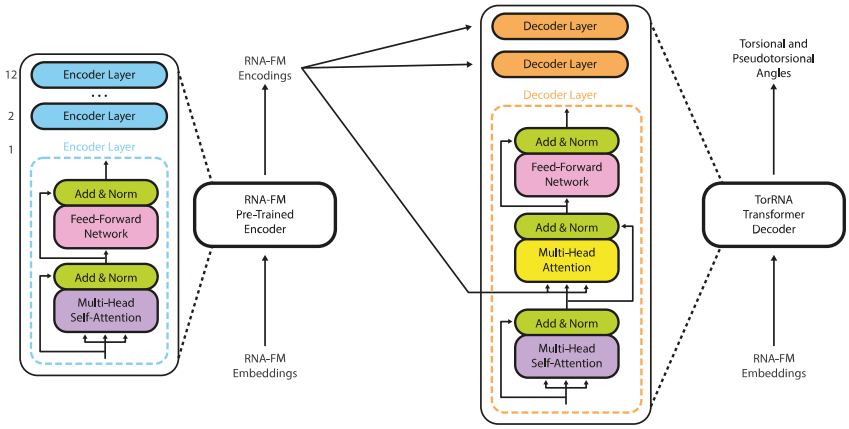
The transformer decoder layers of TorRNA are trained to minimize the Mean Squared Error (MSE) of the *sine* and *cosine* of the (pseudo)torsion angles using the Adam optimizer [94] with the hyperparameters as chosen in Table 3.1. The training and testing of TorRNA was done on a system with a Intel Xeon E5-2640 v4 processor and a GeForce RTX 2080 Ti GPU.

To choose the best hyperparameters for TorRNA's architecture and training procedure, we conduct grid search of the hyperparameters presented in Table 3.1. We chose the best values for the hyperparameters when the error of predicting the (pseudo)torsion angles was the lowest for the RNAs in the validation dataset.

The code and datasets for TorRNA are available at https://github.com/devalab/torrna.



(a) Overall architecture of TorRNA is a transformer encoder-decoder that takes an input RNA sequence and predicts the (pseudo)torsion angles of each nucleotide.



(b) Details of the RNA-FM encoder blocks and the TorRNA decoder blocks that shows how the RNA-FM embeddings are used by the decoder blocks to predict the (pseudo)torsion angles.

Figure 3.2: Overall architecture of TorRNA.

3.3 Results

To evaluate the performance of TorRNA, we use the Mean Absolute Error (MAE), which is the average absolute error between the predicted and ground truth (pseudo)torsion angles. To handle the periodicity of the angles in the MAE calculation, we consider $min(d, 360^\circ - d)$, where d is the absolute difference between two angles. We compare the results of TorRNA with SPOT-RNA-1D [122] and a random predictor. The random predictor works by constructing a histogram of the native angles from the RNAs in the training dataset with a bin-width of 2°, and returns the mean of 100 random predictions using the normalized frequency of each bin as the discrete probability distribution for the center of each bin.

Table 3.1: Search space and the best value for the various hyperparameters for TorRNA

Hyperparameter	Search Space	Best Value
Learning Rate	[0.0001, 0.0002]	0.0002
Hidden Dimension	[256, 512]	256
Number of Attention Heads	[4, 8]	4
Number of Transformer Decoder Layers	[2, 3, 4, 5, 6, 7]	3
Dropout	[0.1, 0.2]	0.2
Tolerance	[3, 5]	5

3.3.1 TorRNA outperforms SPOT-RNA-1D and the random baseline predictor

Table 3.2 and Figure 3.3 compare the performance of TorRNA, SPOT-RNA-1D, and the random baseline predictor in predicting the (pseudo)torsion angles. To provide a direct comparison with SPOT-RNA-1D , we show the performance of TorRNA when trained and tested on dataset splits curated in this thesis in Table 3.2, and on the dataset splits used by SPOT-RNA-1D in Table 3.3.

TorRNA shows improved performance in predicting all torsion angles $(\alpha, \beta, \gamma, \delta, \chi, \epsilon, \zeta)$ and both pseudotorsion angles (η, θ) when compared to both SPOT-RNA-1D and the random baseline predictor. The common trend exhibited by the ML-based prediction methods is that the prediction of the angle delta (δ) has the least average error and the angle alpha (α) has the highest average error. TorRNA and SPOT-RNA-1D have MAEs of 14.26° and 17.1° when predicting the angle delta (δ), and MAEs of 42.1° and 46.1° when predicting the angle alpha (α) . TorRNA predicts the angle delta (δ) with the least error, followed by the angles epsilon (ϵ), chi (χ), beta (β), zeta (ζ), gamma (γ), and alpha (α). When compared to SPOT-RNA-1D, TorRNA achieves an improvement ranging from 2.7% for angle beta (β) to 16.5% for angle delta (δ).

Since the available source code for SPOT-RNA-1D does not allow the model to be retrained with new dataset splits, to obtain a direct comparision, we retrain and test TorRNA on the same RNA molecules on which SPOT-RNA-1D was trained and tested. The performance of the retrained TorRNA and SPOT-RNA-1D are presented in Table 3.3, which show that TorRNA has better predictions of 8/9 of the (pseudo)torsion angles when compared to SPOT-RNA-1D. In the Supplementary Information, we compare TorRNA against other predictors submitted to RNA-Puzzles [133, 134, 135, 136, 137]. TorRNA consistently performs the best in predicting the torsion angles for most puzzles, and gives comparable predictions to the top RNA puzzle predictor in the remaining puzzles.

Table 3.2: MAE of TorRNA compared with SPOT-RNA-1D and the random baseline method for all (pseudo)torsion angles on TorRNA dataset splits

(pseudo)torsion angle	Prediction Method		
	TorRNA	SPOT-RNA-1D	Random Baseline
alpha (α)	**42.052**	46.079	73.044
beta (β)	**20.626**	21.209	123.877
gamma (γ)	**36.443**	37.958	59.064
delta (δ)	**14.257**	17.081	19.538
chi (χ)	**20.11**	21.999	46.129
epsilon (ϵ)	**19.306**	20.311	36.209
zeta (ζ)	**29.182**	30.545	50.646
eta (η)	**25.124**	29.114	79.595
theta (θ)	**28.82**	30.725	67.517

3.3.2 Correlation between TorRNA's prediction errors and (pseudo)torsion angle distributions

The boxplot of the prediction errors of the (pseudo)torsion angles shown in Figure 3.3 shows the distribution of the errors whose averages are presented as the MAEs in Table 3.2. TorRNA's prediction errors follow the same trend as SPOT-RNA-1D where the difficulty of predicting the (pesudo)torsion angles depends on the distribution of the (pseudo)torsion angle. As seen in Figure 3.4, the ground truth values of the angle delta (δ) have a narrow distribution, which explains the low prediction error and the narrow range of the errors in predicting this angle in Figure 3.3. The wide distribution of the ground truth values of the angle alpha (α) explain the prediction errors having a wide range in Figure 3.3 and a high MAE as reported in Table 3.2.

3.3.3 TorRNA's predictive ability for various structural regions of RNA molecules

We investigate TorRNA's (pseudo)torsion angle predictions of nucleotides with various secondary and tertiary interactions with other nucleotides within an RNA molecule. The DSSR [138, 139] software tool marks each nucleotide with the type of interaction in which it is involved. Table 3.4 shows the MAEs obtained by averaging TorRNA's prediction errors for the nucleotides in various structural regions. Figure 3.5 shows the various structural regions that we consider in Table 3.4.

Table 3.3: MAE of TorRNA compared with SPOT-RNA-1D and the random baseline method for all (pseudo)torsion angles on SPOT-RNA-1D dataset splits

(pseudo)torsion angle	Prediction Method		
	TorRNA	SPOT-RNA-1D	Random Baseline
alpha (α)	**38.87**	40.371	72.968
beta (β)	**19.677**	19.82	123.241
gamma (γ)	**31.289**	32.149	55.059
delta (δ)	**12.668**	14.71	17.396
chi (χ)	**16.407**	18.159	48.259
epsilon (ϵ)	19.956	**19.798**	33.564
zeta (ζ)	**27.033**	28.034	49.241
eta (η)	**22.677**	26.537	76.677
theta (θ)	**25.788**	27.887	65.929

The (pseudo)torsion angles of nucleotides that are unpaired ($\sim 28\%$) or are part of hairpin loops ($\sim 12\%$) are the hardest to predict. The difficulty in predicting the (pseudo)torsion angles of these regions could be due to the unpaired nucleotides being very flexible, and TorRNA having no geometric information to infer the nucleotides in hairpin loops which have a distribution of angles away from the remaining nucleotides. Nucleotides that are a part of canonical nested pairs make up $\sim 47\%$ of all nucleotides and are the easiest to predict. As seen in Table 3.4, TorRNA predicts all (pseudo)torsion angles better than SPOT-RNA-1D across most structural region, and gives comparable results to SPOT-RNA-1D in the remaining cases.

3.3.4 TorRNA's robustness to the length of RNA sequences

The lengths of the longest RNA sequence in the training, validation, and test sets varying greatly could potentially affect the performance of TorRNA on long RNA molecules. To analyse this, Figure 3.6 shows the MAEs of all (pseudo)torsion angles for RNA molecules of varying sequence lengths. All the (pseudo)torsion angles largely have the same MAEs for RNAs of all lengths. It can also be noted that there is no clear loss in performance in predicting the (pseudo)torsion angles of RNAs of greater lengths, with some angles even having their lowest prediction errors for the longest RNAs in the test dataset.

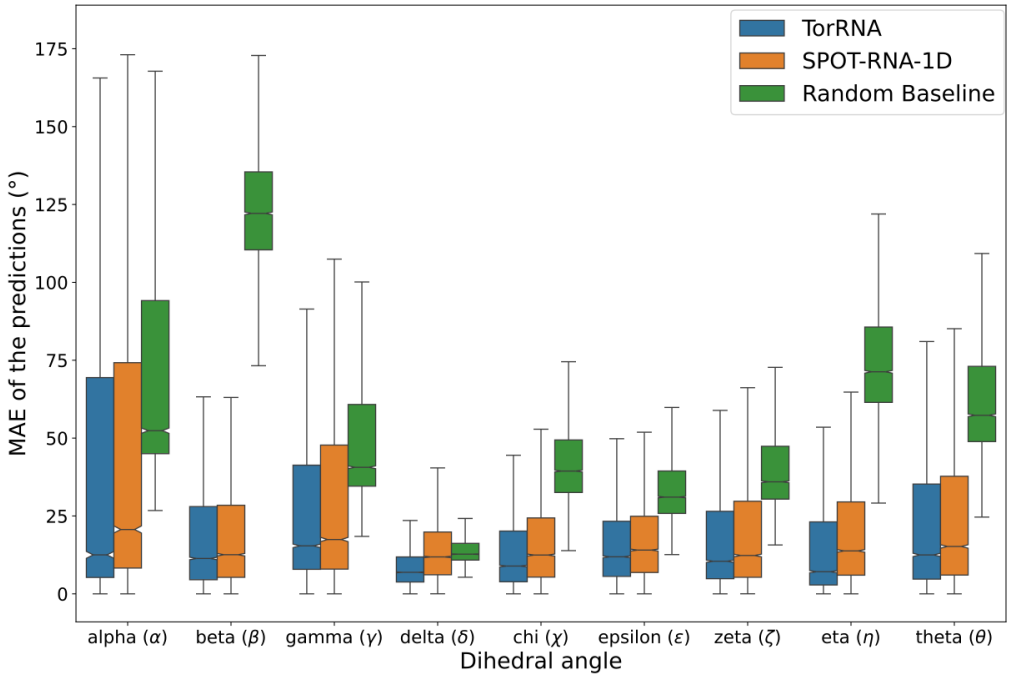


Figure 3.3: Boxplot of the prediction errors of the (pseudo)torsion angles to compare the distribution of the errors of TorRNA, SPOT-RNA-1D, and the random baseline predictor.

3.3.5 Using TorRNA as a model evaluator

While developing Ribonucleic Acids Statistical Potential (RASP) [142] - an all-atom knowledge-based potential for the assessment of 3D RNA structures - the authors use 500 decoy models for each of the 85 native RNA structures in a dataset that they name *randstr* decoy set to test the knowledge-based potential they developed. These decoys were built with the MODELLER computer program [143] using an increasingly smaller subset of Gaussian potentials as restraints on the dihedral angles and atomic distances.

Out of these 85 RNAs, 2 RNAs are present in the testing dataset of TorRNA and are non-redundant with the training dataset. We use these 2 RNAs to explore the connection between the prediction errors of TorRNA and the structural accuracy of the models measured by the root-mean-square deviation (RMSD) and global distance test (GDT) score [144] to their native structures. Figure 3.7 plots the MAEs between the (pseudo)torsion angles predicted by TorRNA and the angles of the decoy models against the structural accuracy of the models for the PDB IDs 1MZP (Chain B) and 387D (Chain A). The MAE of the predictions increase as the structural accuracy of the models decrease, i.e. as the RMSD increases

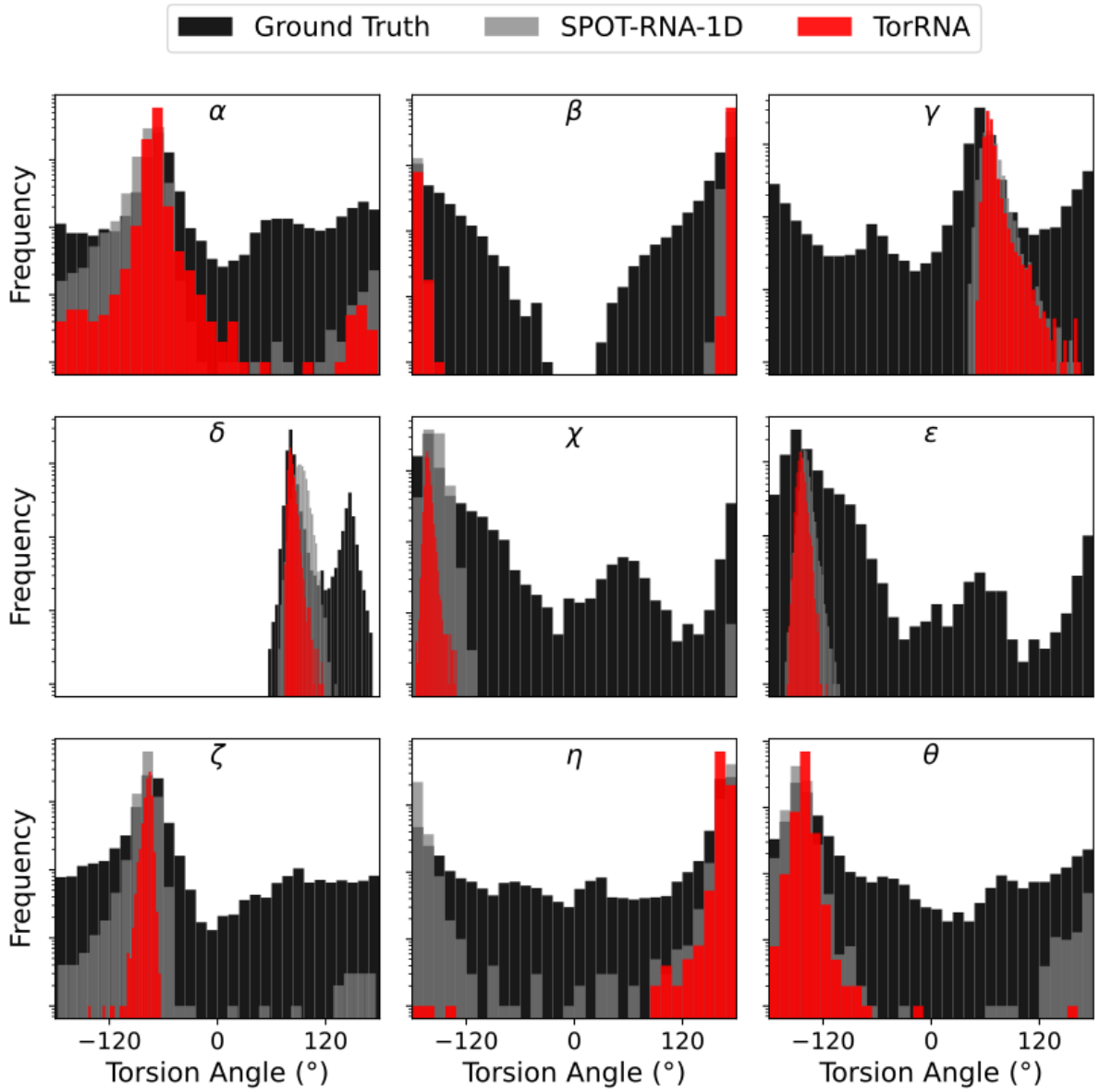
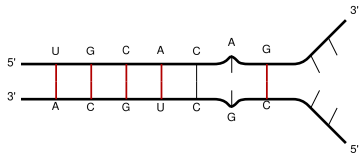
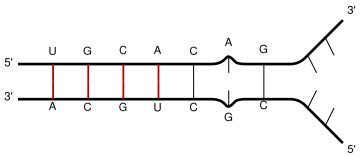


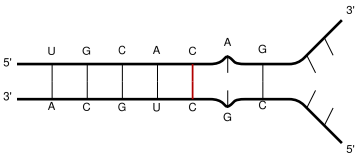
Figure 3.4: Histograms of ground truth (pseudo)torsion angles and those predicted by TorRNA and SPOT-RNA-1D. The Y-axis uses a logarithmic scale to show the frequency of each frequency bin in the histogram.



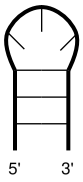
(a) Canonical Base Paired Residues.



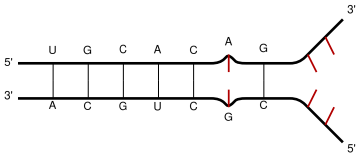
(b) Canonical Nested Base Paired Residues.



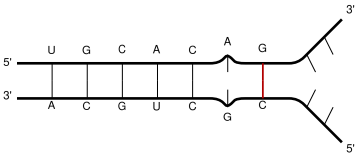
(c) Non-Canonical Base Paired Residues.



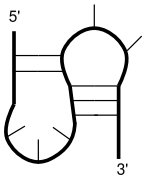
(d) Hairpin Loop Residues.



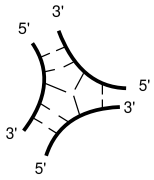
(e) Unpaired Residues.



(f) Lone Base Paired Residues.



(g) Pseudoknot Residues.



(h) Multiplet Residues.

Figure 3.5: The various structural regions of RNA molecules that we consider. The specific residues are highlighted in red when the region is ambiguous from the figure.

Table 3.4: MAE of angles predicted by TorRNA in various regions of an RNA molecule with the MAE of the predictions by SPOT-RNA-1D in the parenthesis. SPOT-RNA-1D MAEs are in bold when they are lower than the corresponding MAE of TorRNA.

Region (% of all nucleotides present in this region)	alpha (α)	beta (β)	gamma (γ)	delta (δ)	chi (χ)	epsilon (ϵ)	zeta (ζ)	eta (η)	theta (θ)
All Canonical Pairs (50.51%)	29.51 (32.72)	15.06 (15.66)	27.54 (28.26)	7.82 (11.66)	9.37 (12.15)	13.80 (15.44)	15.18 (16.28)	11.10 (15.48)	14.43 (16.05)
Canonical Nested Pairs (46.49%)	33.32 (36.56)	16.89 (17.33)	30.45 (31.25)	9.01 (12.41)	12.43 (14.54)	15.13 (16.73)	18.51 (19.47)	13.95 (17.94)	17.30 (18.79)
Non-Canonical Pairs (25.61%)	47.24 (50.10)	24.14 (24.64)	41.32 (41.66)	15.58 (17.91)	24.48 (25.48)	22.42 (22.66)	37.94 (38.76)	25.94 (28.14)	33.89 (34.86)
Hairpin Loops (11.72%)	62.57 (62.66)	26.55 (27.14)	42.77 (42.92)	22.75 (23.57)	28.52 (**28.47**)	26.57 (**26.55**)	46.18 (46.69)	48.98 (51.52)	45.43 (46.73)
Unpaired (27.77%)	61.56 (63.52)	29.32 (29.97)	48.21 (48.91)	22.45 (23.69)	32.14 (32.58)	26.75 (26.79)	46.76 (47.87)	48.18 (50.83)	48.83 (50.05)
Lone Pairs (5.99%)	44.43 (46.32)	22.18 (23.19)	37.18 (37.55)	16.28 (18.67)	22.48 (23.55)	22.96 (**22.73**)	42.93 (43.84)	25.10 (27.46)	36.16 (36.57)
Pseudoknots (2.71%)	38.30 (40.46)	18.51 (18.61)	30.23 (31.47)	11.88 (13.76)	13.38 (14.58)	18.73 (18.80)	27.96 (28.78)	23.60 (25.31)	27.61 (28.30)
Multiplets (9.24%)	50.89 (53.55)	24.89 (25.33)	42.53 (42.99)	17.36 (18.81)	25.07 (25.29)	22.88 (**22.78**)	42.60 (42.76)	27.60 (28.56)	37.04 (37.53)

and the GDT decreases. This shows that the MAE of the predictions can serve as an effective proxy when the RMSD and GDT scores are not available, which is the case when generating the structure of a novel RNA.

In Figure 3.8a, we plot the distribution of the MAEs between the (pseudo)torsion angles predicted by TorRNA and the angles of the decoy models for the decoy models that have the minimum and maximum MAE for each RNA in the *randstr* decoy set. Figure 3.8b plots the distributions of the RMSDs of decoy models that have the minimum and maximum MAEs against the angles predicted by TorRNA. Both these figures show that the MAEs and RMSDs of the decoy models with minimum and maximum MAEs show disjoint distributions, implying that the MAE calculated against the (pseudo)torsion angles by TorRNA is a good metric to assess the quality of the decoy models. Figure 3.9 shows the best (green) and worst (red) decoy models against the native structure (black) of 3 RNAs.

These results show that the difference of the (pseudo)torsion angles predicted by TorRNA from the angles of a candidate model structure could be used as a model quality assessment of the candidate 3D structure of the RNA molecule. TorRNA's MAEs can be used to distinguish and correctly rank candidate models of RNA structures, even when the candidate models have minimal structural deviation. TorRNA can work as a powerful RNA model quality assessment tool to rank candidate models generated by ML-based methods or through other methods.

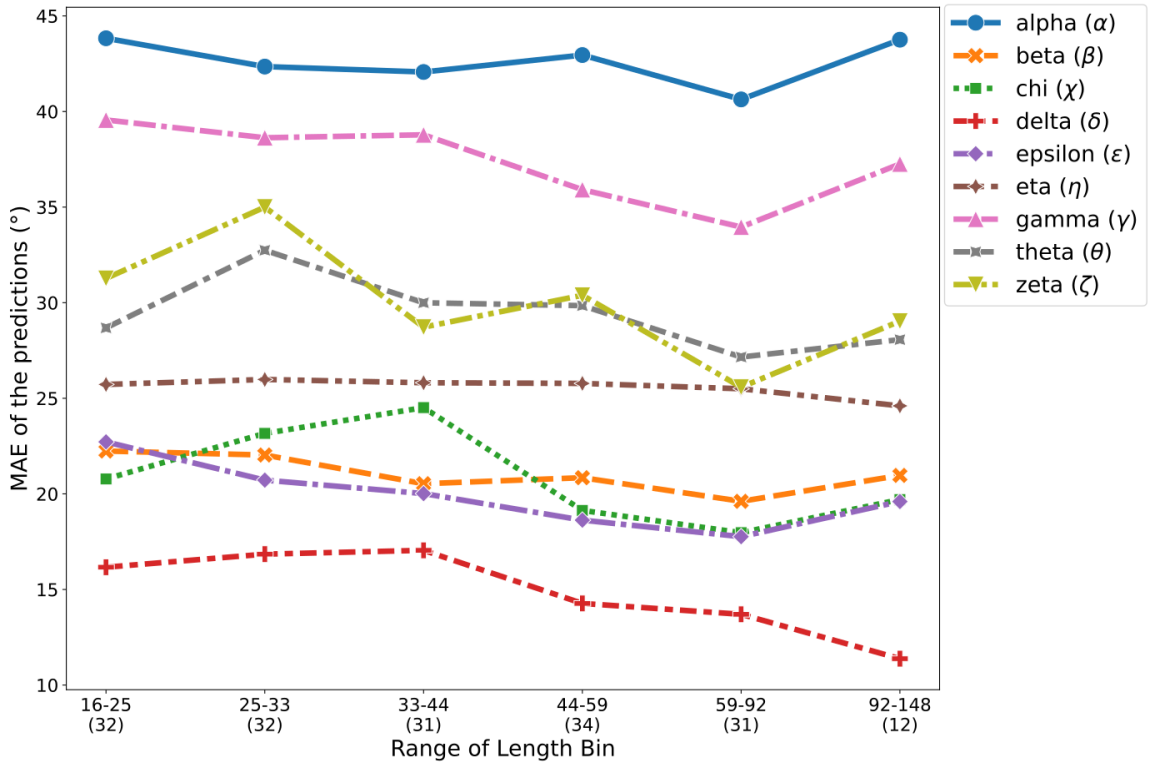


Figure 3.6: MAEs of the (pseudo)torsion angles for various RNA sequence lengths. The X-axis labels describe the length bins along with the number of RNAs that are in each length bin.

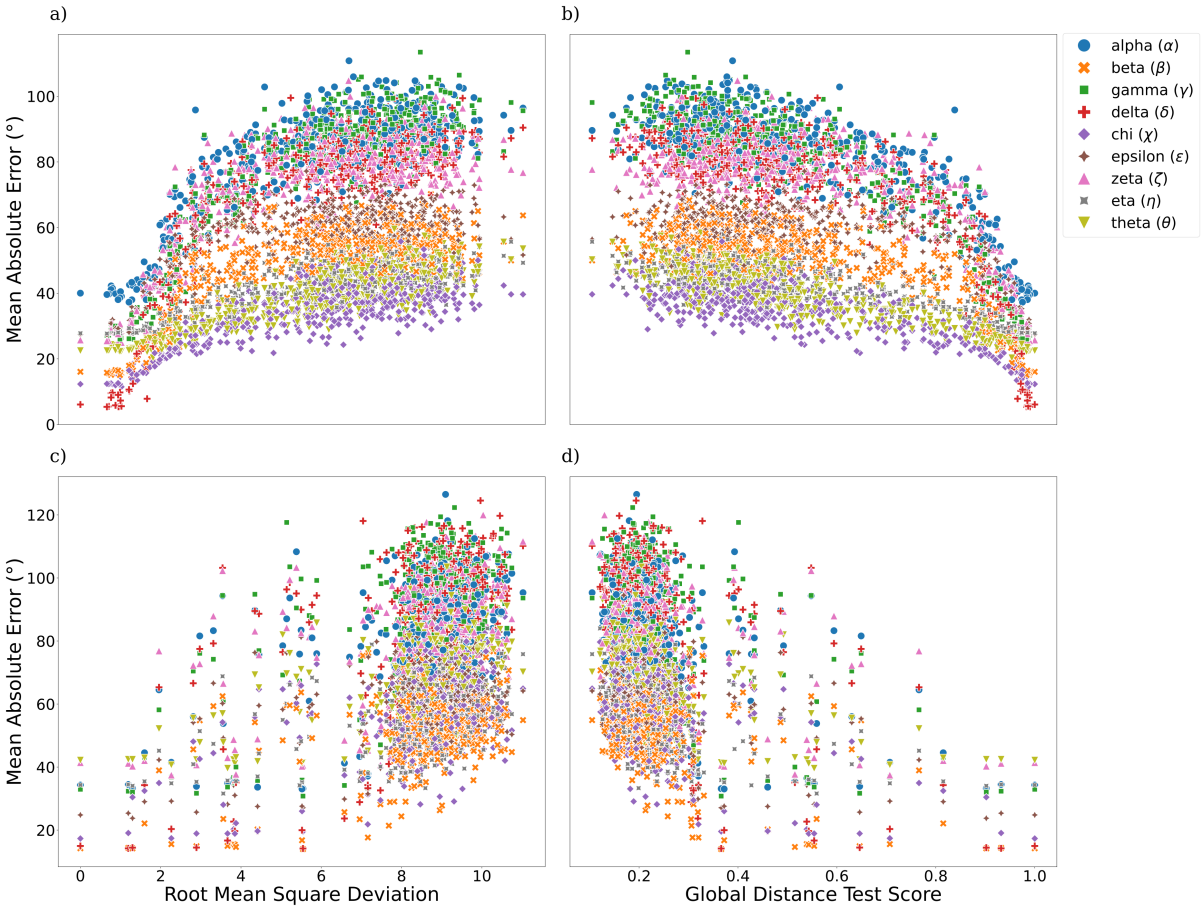


Figure 3.7: MAE vs RMSD and MAE vs GDT scatterplots for PDB ID 1MZP (Chain B) (a, b) and 387D (Chain A) (c, d)

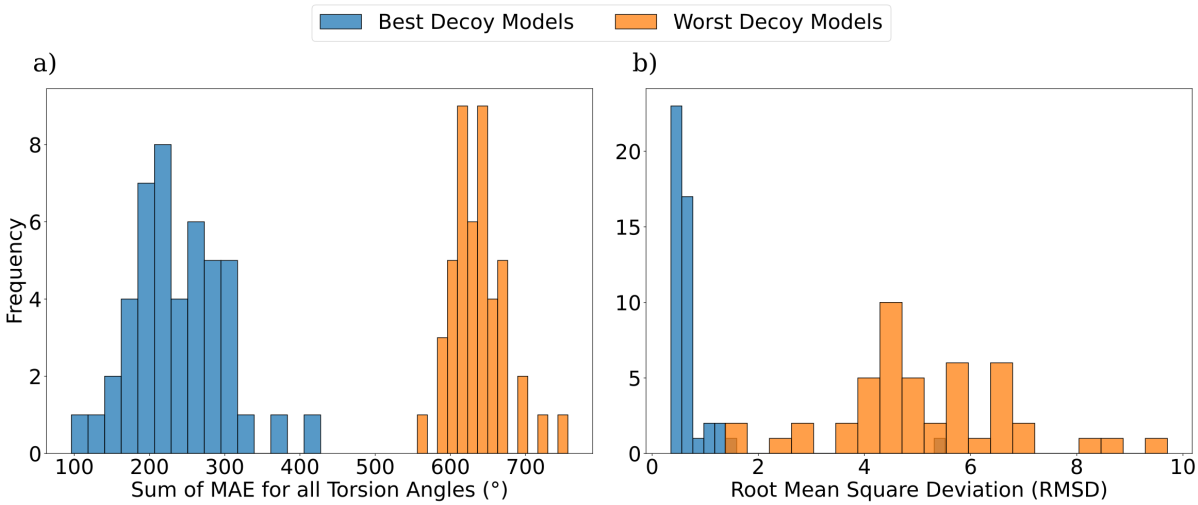


Figure 3.8: The MAE of a model's angles against TorRNA's predictions separates the best and worst decoy models both in terms of the MAE, and also in terms of the RMSD of the decoy structures with the native structure.

(a) Best model for PDB ID 1Z43 (Chain A) according to TorRNA. RMSD - 0.405 $\mathring{A}$; MAE - 284°.

(b) Worst model for PDB ID 1Z43 (Chain A) according to TorRNA. RMSD - 4.582 $\mathring{A}$; MAE - 608°.

(c) Best model for PDB ID 3CPW (Chain 9) according to TorRNA. RMSD - 0.687 $\mathring{A}$; MAE - 207°.

(d) Worst model for PDB ID 3CPW (Chain 9) according to TorRNA. RMSD - 4.626 $\mathring{A}$; MAE - 583°.

(e) Best model for PDB ID 3F1H (Chain B) according to TorRNA. RMSD - 0.605 $\mathring{A}$; MAE - 169°.

(f) Worst model for PDB ID 3F1H (Chain B) according to TorRNA. RMSD - 4.667 $\mathring{A}$; MAE - 622°.

Figure 3.9: The native structure (black) of various RNAs and the decoy model with the lowest (green) and highest MAE (red) against the angles predicted by TorRNA to show TorRNA's potential to be used as a model quality assessment tool. The caption of each subfigure also contains the RMSD of the decoy model to the native structure, and the sum of MAE between TorRNA's predictions and the decoy model's (pseudo)torsion angles.

Chapter 4

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.

TorRNA is a transformer encoder-decoder model, that takes an input RNA sequence and predicts the (pseudo)torsion angles of each nucleotide with a pre-trained RNA-FM model as the transformer encoder. Since the secondary structure being predicted are the (pseudo)torsion angles, TorRNA is able to employ a transformer decoder that takes the encodings from a pre-trained transformer encoder. This sets

TorRNA apart from other works that use a CNN-based architecture to predict the secondary structure of proteins and nucleic acids from encodings derived by foundation models. TorRNA also curates new dataset splits of the RNAs that have high-resolution 3D structures available, to take into the account new data that might have been gathered since the previous (pseudo)torsion angle prediction method was released.

TorRNA is able to achieve a performance boost of $2\% - 16\%$ over the previous (pseudo)torsion angle prediction method SPOT-RNA-1D and consequently shows an improved performance over a random baseline predictor as well. TorRNA is also robust in terms of predicting the (pseudo)torsion angles for RNAs of various sizes, and for nucleotides in various structural regions of the RNA molecules. With this improved prediction of the (pseudo)torsion angles, these predictions can be used as restraints on the dihedrals for the optimization of unrefined RNA structures. We also demonstrate the potential of TorRNA to be used as a tool for model quality assessment of candidate RNA structures for a given RNA sequence.

We believe that TorRNA is a valuable contribution that would help spur further research in improving sequence to structure methods for RNA molecules and take a step towards unleashing the therapeutic value of RNA molecules to develop better drugs.

This thesis removes the database requirement for algorithms that depend on database searches for structure prediction in two areas within computational chemistry - molecular structure elucidation from molecular spectra, and tertiary structure prediction of RNA (Ribonucleic acid) molecules from their sequence by proposing the methods DeepSPInN and TorRNA. This thesis makes progress in addressing the problem of a lack of diversity within individual databases, and makes these algorithms more accessible to researchers who are unable to access or have their own copy of enormous databases. Hopefully, this thesis brings attention to the inaccessibility and bias for algorithms that depend on database searches, and spurs further work into developing machine learning algorithms that learn from the databases but do not depend on the database search algorithms.

Appendix A

Related Publications

1. **Devata, Sriram**, Sridharan, B., Mehta, S., Pathak, Y., Laghuvarapu, S., Varma, G., & Priyakumar, U. D. (2024). DeepSPInN – deep reinforcement learning for molecular structure prediction from infrared and 13C NMR spectra. Digital Discovery, 3, 818–829. doi:10.1039/D4DD00008K

2. **Devata, Sriram** and Priyakumar, U. D. (2024). "TorRNA - Improved Prediction of Backbone Torsion Angles of RNA by Leveraging Large Language Models." ChemRxiv. doi:10.26434/chemrxiv-2024-cj4r0 This content is a preprint and has not been peer-reviewed.

Appendix B

Supplementary Information for DeepSPInN

B.1 Statistics for the dataset

To understand the distribution of the molecules in the dataset, Figures B.1 and B.2 show the distribution of the molecular weights and the counts of molecules that have the heavy atoms C, N, O, and F.

B.2 Congruence of simulated and experimental Infrared spectra

Of the 40 molecules whose infrared spectra we were able to download from the NIST Quantitative Infrared Database, we had the simulated spectra of 15 molecules. The average SIS of the experimental and simulate infrared spectra is 0.2241. Although predictions are expected to have SIS in the range 0.40–0.70 to even be considered as loosely predictive, the simulated spectra should not be considered as replacements for experimental spectra.

B.3 Threshold reward for MCTS

MCTS will continue finding child nodes to the search tree until it reaches a terminal state. For a state to be considered a terminal, at least one of these following termination criteria have to be met:

1. There are no more valid actions
2. The reward of this state is greater than a particular threshold
3. The NMR split values of the current state are such that the target NMR split values can never be reached

If there are no valid actions that can be taken from a state, it has to be terminal since there are no possible child nodes. If a state has a molecular graph where there are no individual nodes, further actions

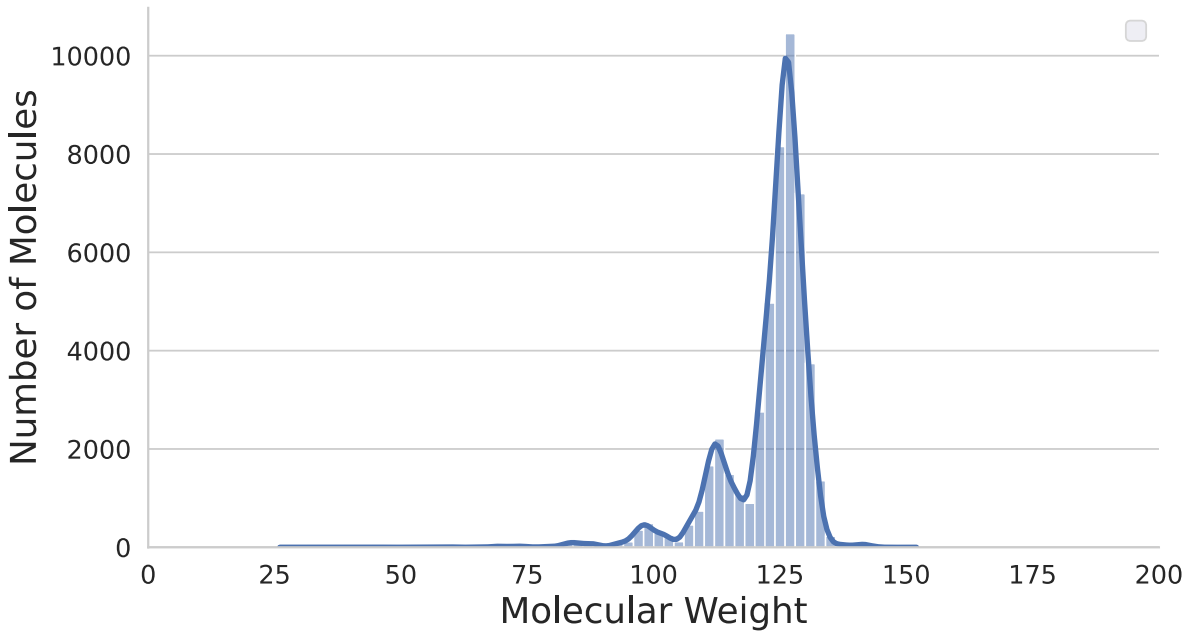


Figure B.1: Distribution of the molecular weights in the dataset

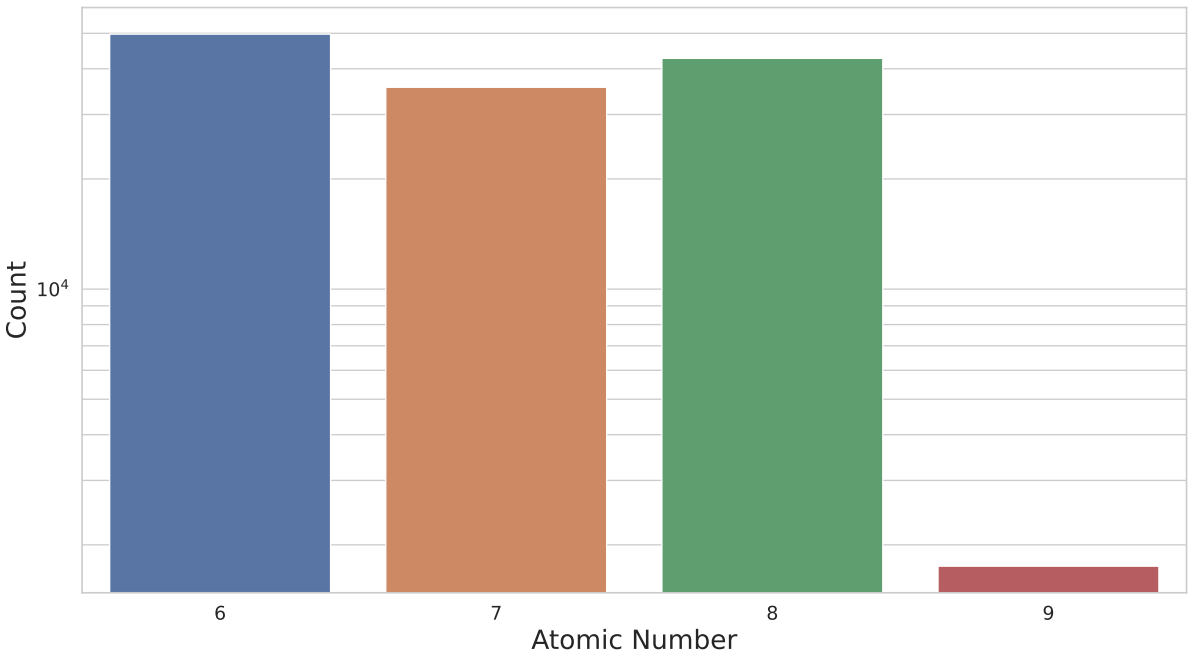


Figure B.2: Number of molecules that contain each element

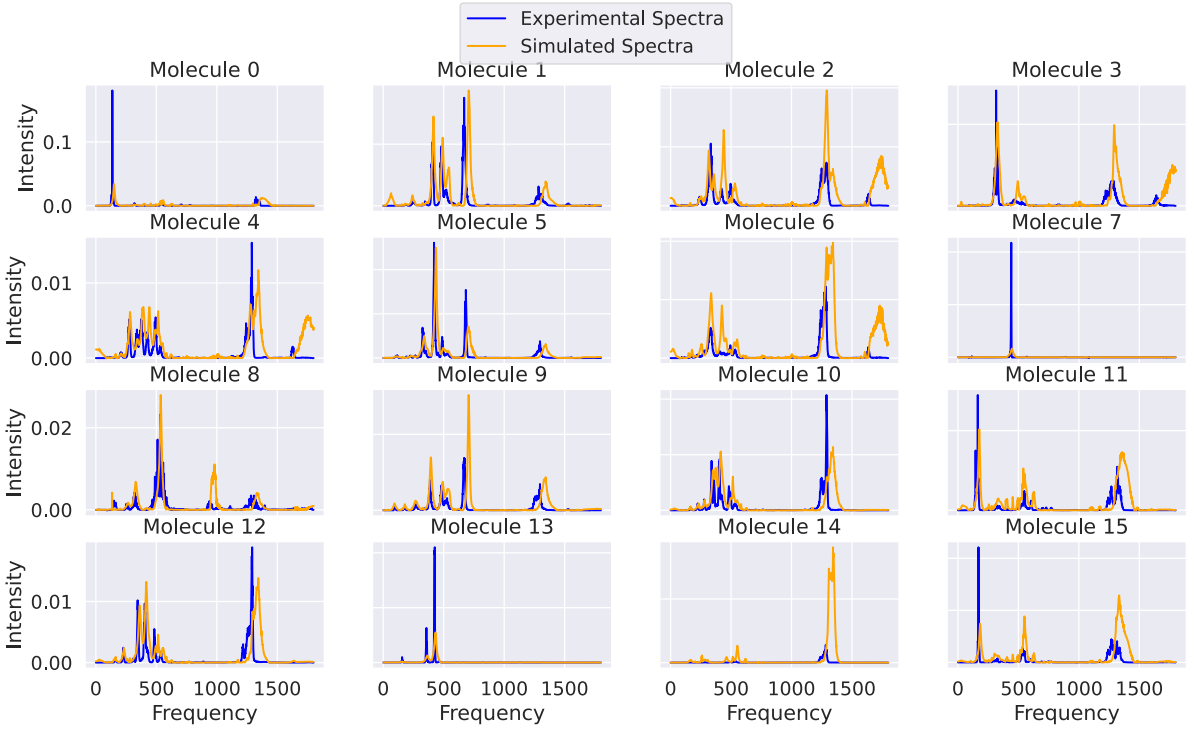


Figure B.3: Experimental and Simulated IR spectra

will just continue adding bonds within the molecule. In such a case, the environment checks if the NMR split values of a state can still allow further addition of bonds. If a state's NMR split values are such that the target NMR split values cannot be reached via the addition of more bonds, then the state is considered to be terminal.

In addition to these termination criteria, we want the framework to stop the tree search when it is confident that the target molecule has been reached. We use the reward function to get the reward for each state, and if the reward is greater than a particular threshold, the state is terminal. To choose this threshold value, we sampled 10,000 molecules from the test set and found the SIS between their IR spectrum and the predicted IR spectrum from Chemprop-IR. These rewards are plotted in Fig. B.4. 88.63% of molecules have their self-rewards above 0.95. We sampled another 100 molecules and for each molecule, we found the SIS between the molecule's IR spectrum and the Chemprop-IR predicted spectrum of the other 99 molecules. These rewards are plotted in Fig. B.5. All these rewards are below 0.95. Choosing 0.95 as a reward threshold means that states with a reward greater than this threshold are very likely to be the target molecules themselves. This allows the framework to stop the tree search at this point.

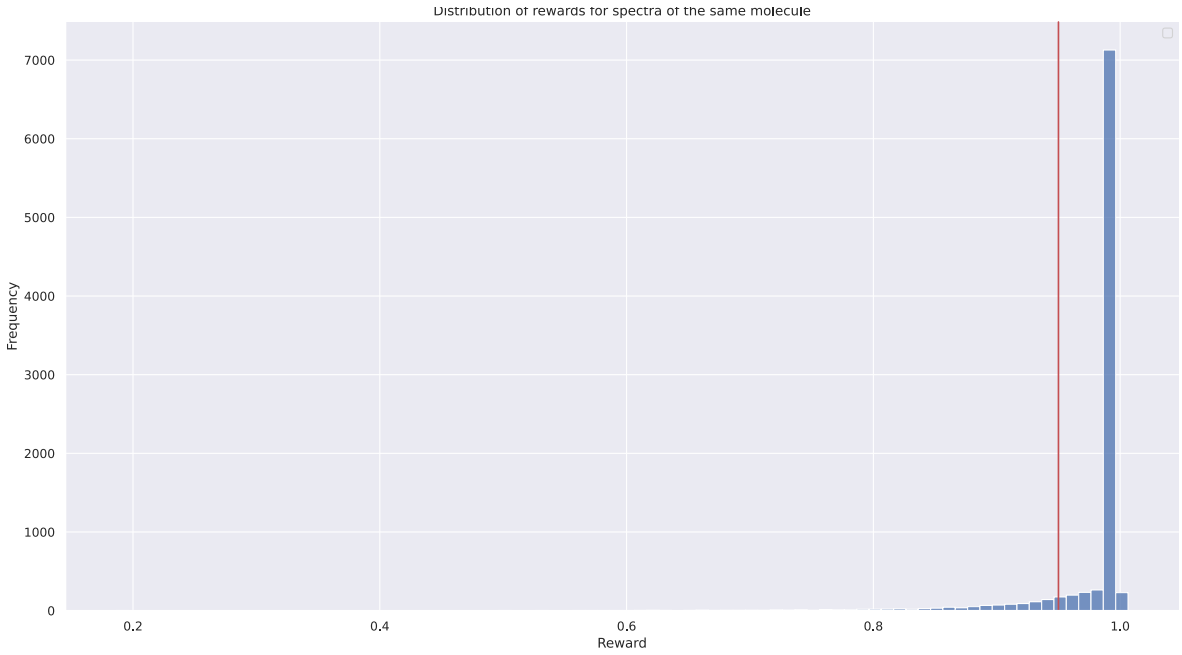


Figure B.4: Distribution of rewards of the same molecules

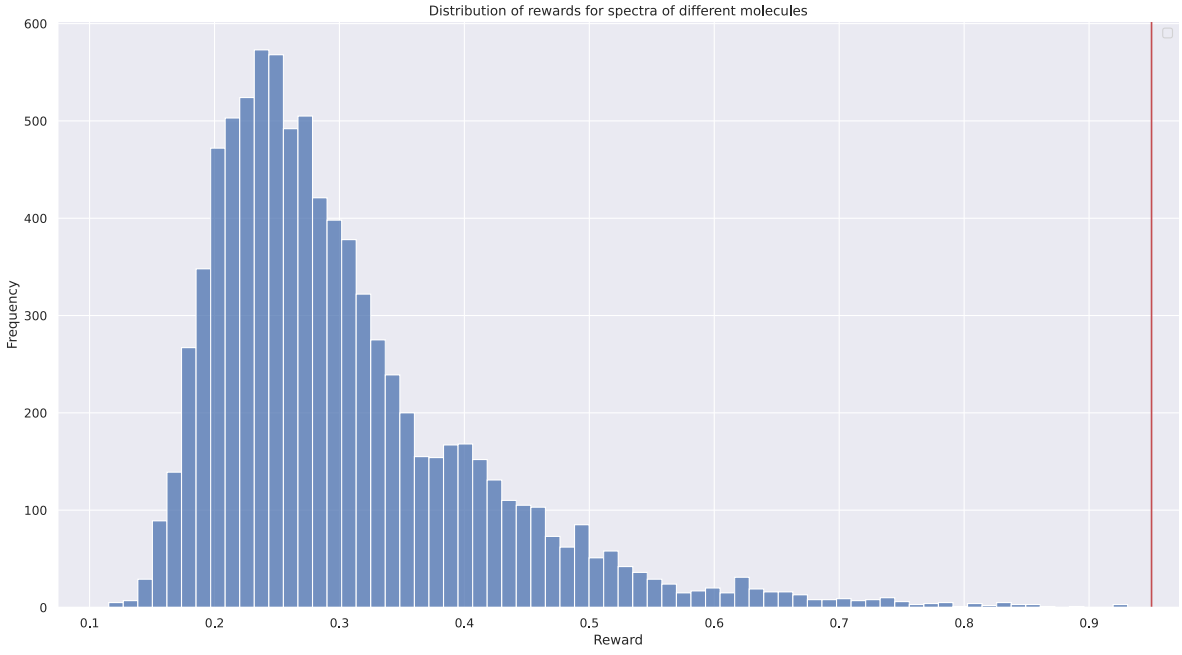


Figure B.5: Distribution of rewards between different molecules

B.4 SIS Loss

The original SIS description performs a Gaussian convolution to allow for any minor deviations in the spectral peak locations. This Gaussian convolution allows spectra with minor differences in the peak locations and intensities to have a relatively high SIS score. We do not perform this Gaussian convolution while calculating SIS since the dataset used in this work already has broadened stick spectra and Chemprop-IR also predicts broadened IR spectra.

B.5 Training on molecules with ≤ 7 heavy atoms and testing on molecules with 8 or 9 heavy atoms

For the generalization study where the framework was trained on molecules with up to 7 heavy atoms, and tested on molecules with 8 or 9 heavy atoms, the test-train split of the dataset was heavily unbalanced. There are a total of 2,099 molecules with ≤ 7 heavy atoms, with all these molecules making the training set. The remaining 47,650 molecules have either 8 or 9 heavy atoms and these molecules make up the testing set. Due to computational constraints of testing the framework on the entire set, the framework was tested on a random subset of 5000 molecules that contain either 8 or 9 heavy atoms. Out of these 5000 molecules, 771 molecules have 8 heavy atoms and 4,229 have 9 heavy atoms.

B.6 Choosing the number of episodes hyperparameter

Due to the way we parallelized DeepSPInN and with the resources that we used, it takes the same amount of time to run the number of episodes that is the next closest multiple of 8. For example, it takes the same amount of time to run 22 episodes and 24 episodes. This is why we chose 32 episodes even though running for 28 episodes has a negligible decrease in the fraction of correctly predicted molecules when compared to 32.

B.7 *Top N* metrics for various functional groups/structural motifs

To identify if DeepSPInN has any affinity to predict the molecular structures that contain specific functional groups better than others, we analyzed the *Top N* metrics of the molecules that contain different functional groups. DeepSPInN performs well for molecules with ketones, with a *Top 1 (%)* accuracy of 96.24%, and performs the worst for molecules with amines with a *Top 1 (%)* accuracy of 86.99%. Table B.1 shows the *Top N (%)* metrics of other functional groups as well as the number of molecules in the test set that contained these functional groups.

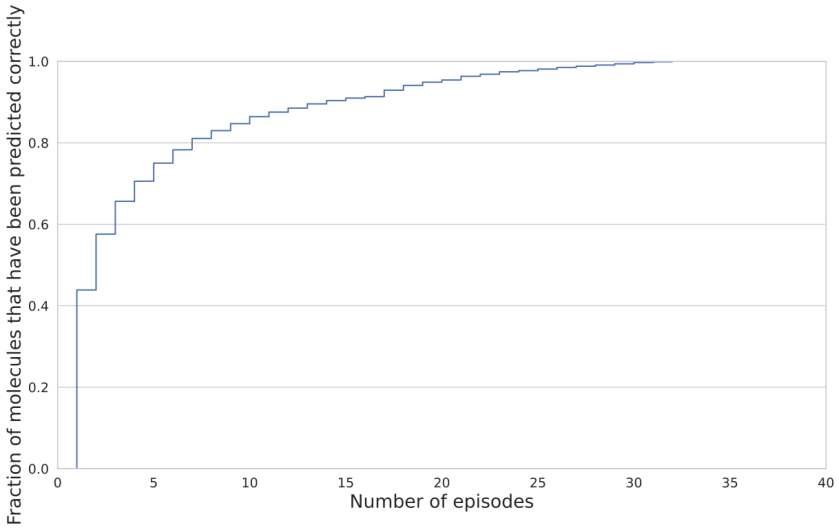


Figure B.6: Cumulative plot of the fraction of correctly predicted molecules and the number of episodes that were taken to find the right molecule

Table B.1: *Top N* metrics for molecules that have specific functional groups

Functional Group (Number of molecules)	Alcohols (2987)	Aldehydes (1214)	Amines (1983)	Ester (437)	Ketone (1064)	Phenol (556)
Top 1 (%)	92.37	93.49	86.99	93.82	96.24	90.83
Top 3 (%)	93.34	94.48	87.59	94.51	97.27	91.01
Top 5 (%)	93.40	94.65	87.64	94.51	97.37	91.01
Top 10 (%)	93.44	94.65	87.70	94.51	97.37	91.01

B.8 Using proton-coupled ^{13}C NMR spectra

In an experiment where we assume that we also have the ^{13}C NMR split values, we use these values to help prune the search tree by identifying molecules that are invalid according to the input NMR spectrum and nullify the rewards for these molecules. This discourages the tree search from exploring these molecules. The NMR split values for the NMR shift of each carbon atom are equivalent to the number of hydrogen atoms attached to it. Each carbon atom can be either a singlet (S, quarternary), doublet (D, tertiary), triplet (T, secondary), or a quartet (Q, primary) atom with each of these denoted by S, D, T, and Q respectively. Since each valid action is defined as the addition of a bond between two atoms in the MDP reformulation, each valid action can only convert the carbon atoms from $Q \rightarrow T \rightarrow D \rightarrow S$. If the target Q-splits are more than the Q-splits at one such state, this state is not valuable

Table B.2: *Top N* metrics for varying n_{mcts} values

	IR+NMR			
n_{mcts}	100	200	400	800
Top 1 (%)	82.311	90.773	94.934	95.980
Top 3 (%)	82.874	91.597	95.839	96.925
Top 5 (%)	82.994	91.778	96.060	97.106
Top 10 (%)	83.015	91.798	96.120	97.166
Top 40 (%)	83.015	91.798	96.120	97.206

Table B.3: Performance of IR-and-NMR-trained, IR-trained, and NMR-trained models for $n_{mcts} = 400$

	IR and NMR	Only IR	Only NMR
Top 1 (%)	94.934	74.075	40.462
Top 3 (%)	95.839	74.396	61.849
Top 5 (%)	96.060	74.396	68.201
Top 10 (%)	96.120	74.396	73.105
Top 40 (%)	96.120	74.423	74.472

since no valid action from this state would be able to increase the count of Q-splits. If the Q-splits match, checking the T and D-splits subsequently in the same way further identify more states that get zero-rewards.

Tables B.2, B.3, and B.4 present the same results from the main paper, but with the ^{13}C NMR split values being used. The dataset split is different from the main paper, with a 80-20 split being used for the train and test sets.

B.9 Testing DeepSPInN checkpoints trained on simulated spectra to elucidate experimental spectra

The simulated Infrared and 13C NMR spectra used to train and test DeepSPInN reflect the complexity of experimental spectra but can not serve as replacement for the experimental spectra. Since DeepSPInN is able to work with simulated spectra, it can analogously learn to work with experimental spectra.

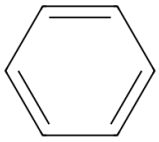
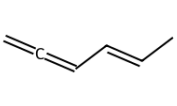
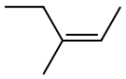
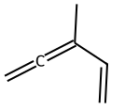
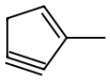
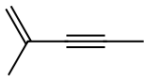
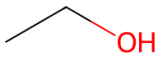
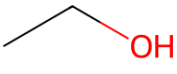
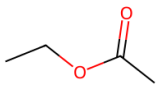
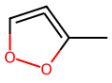
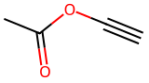
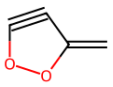
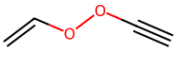
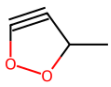
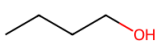
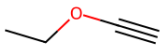
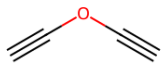
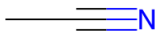
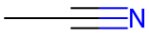
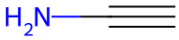
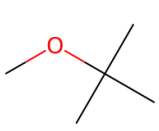
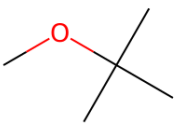
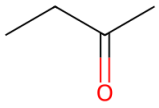
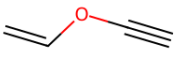
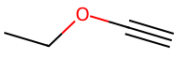
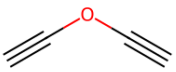
Table B.4: Training on molecules with ≤ 7 atoms and testing on molecules with ≥ 8 atoms for $n_{\text{mcts}} = 400$

	≥ 8 atom molecules	8-atom molecules	9-atom molecules
Top 1 (%)	80.971	96.024	77.948
Top 3 (%)	82.046	97.247	78.992
Top 5 (%)	82.148	97.247	79.115
Top 10 (%)	82.199	97.247	79.176
Top 40 (%)	82.250	97.247	79.238

With this thesis focusing on the development of DeepSPInN as a proof of concept, the current DeepSPInN model checkpoints can not (should not) be used for testing on experimental data since it was only trained on simulated data. DeepSPInN is able to learn the complexity of spectra, as seen by its performance on simulated spectra, and would perform well on unseen experimental spectra when it is also trained on experimental spectra. Future works would be able to construct databases of experimental Infrared and ^{13}C NMR spectra to train checkpoints of DeepSPInN that would work well on new experimental data.

We demonstrate that the current DeepSPInN checkpoints do not perform well on elucidating the structures of experimental spectra by gathering the experimental Infrared and ^{13}C NMR spectra of 14 molecules from the databases NIST Quantitative Infrared Database and nmrshiftdb2. In Table B.5, we show the top candidate molecules of some of these molecules as predicted by DeepSPInN. As expected, DeepSPInN did not perform well and was only able to resolve 3/14 of the molecules.

Table B.5: Top candidate molecules and the rewards when given experimental IR and ^{13}C NMR spectra as input to a DeepSPInN model trained on simulated spectra

Target Molecule	Top 5 Candidate Molecules (and their final rewards)				
 Target Molecule	 Reward: 0.133	 Reward: 0.127	 Reward: 0.127	 Reward: 0.123	 Reward: 0.123
 Target Molecule	 Reward: 0.270				
 Target Molecule	 Reward: 0.289	 Reward: 0.273	 Reward: 0.233	 Reward: 0.214	 Reward: 0.192
 Target Molecule	 Reward: 0.233	 Reward: 0.231	 Reward: 0.190	 Reward: 0.107	
 Target Molecule	 Reward: 0.277	 Reward: 0.125			
 Target Molecule	 Reward: 0.242				
 Target Molecule	 Reward: 0.204	 Reward: 0.200	 Reward: 0.189	 Reward: 0.087	

Appendix C

Supplementary Information for TorRNA

C.1 Comparison of TorRNA with the top RNA Puzzles submissions

While making the dataset splits, all the RNA molecules that were a part of the RNA-Puzzles were maintained in TorRNA's test dataset. With these RNAs being non-redundant with the training dataset of TorRNA, we calculated the MAEs of the (pseudo)torsion angles between the puzzle's solution and submissions for 9 RNA puzzles. We compare the closeness of the angles of the 3D structures given by the predictors for these RNA puzzles and the angles predicted by TorRNA to their experimentally determined native structures in Table C.1. TorRNA consistently performs the best in predicting the torsion angles for most puzzles, and gives comparable predictions to the top RNA puzzle predictor in the remaining puzzles.

Table C.1: MAE of (pseudo)torsion angles from structures predicted by submissions to the RNA Puzzles competition when compared to TorRNA's predicted (pseudo)torsion angles

Submission	alpha (α)	beta (β)	gamma (γ)	delta (δ)	chi (χ)	epsilon (ϵ)	zeta (ζ)	eta (η)	theta (θ)
rp19									
TorRNA	**31.50**	**12.79**	**23.72**	13.34	**12.71**	**11.38**	**16.47**	19.68	25.57
19_RW3D	45.63	19.24	30.84	13.93	16.98	19.72	26.17	28.59	33.63
19_Das_Human	39.07	23.32	30.23	**11.05**	12.81	19.99	25.18	20.19	29.86
19_SimRNA	50.34	21.96	33.75	15.13	19.00	22.39	44.70	33.98	40.79
19_Ding_Human	37.55	18.05	28.31	16.77	15.15	13.41	20.80	23.42	31.44
19_3dRNA	64.21	37.15	53.03	30.67	18.18	38.71	42.04	33.17	38.45
19_Dokholyan	49.47	20.29	51.30	14.85	18.40	12.67	23.71	29.31	34.16
19_LeeServer	34.73	17.87	28.32	25.31	24.20	29.19	24.57	25.22	33.43
19_Chen_Human	46.04	26.70	31.61	12.76	19.13	31.90	40.17	20.00	**22.34**
19_RNAComposer_Human	42.44	26.11	36.06	12.56	18.59	17.17	32.34	21.37	38.04
19_RNAComposer	44.57	27.30	38.11	13.70	18.07	16.53	32.79	**19.53**	35.78
19_Bujnicki_Human	39.57	18.03	36.00	16.88	15.30	18.23	20.90	22.64	27.44
rp21									
TorRNA	**53.03**	**22.00**	**41.33**	18.02	**20.91**	**18.68**	**37.12**	**19.77**	35.66
21_Das	61.72	31.70	53.55	**15.39**	20.99	23.91	41.21	26.72	**28.30**
21_Sanbonmatsu	61.46	45.81	56.49	29.20	29.01	42.09	45.34	31.58	55.01
21_RNAComposer	68.03	36.22	65.71	22.83	32.43	28.24	51.74	36.60	48.72
21_ChenHighLig	57.25	24.79	43.52	19.22	25.20	28.32	43.28	25.30	38.02
21_Bujnicki	65.21	27.26	46.95	24.02	25.26	24.38	42.75	29.74	40.39
21_RW3D	70.28	29.41	54.25	18.59	23.43	28.03	46.50	35.31	47.68
21_ChenLowLig	57.27	24.77	43.52	19.21	25.19	28.34	43.26	25.30	38.02
21_simRNA	58.74	27.14	55.74	18.88	22.48	26.36	46.95	32.88	44.75
21_DasLORES	61.72	31.70	53.55	**15.39**	20.99	23.91	41.21	26.72	**28.30**
21_Adamiak	74.02	25.64	58.76	21.00	25.06	27.79	50.23	32.42	43.39
rp15									
TorRNA	**22.04**	**10.48**	**17.74**	11.64	**10.71**	**11.58**	23.83	20.01	23.71
15_RNAComposerAS1	25.71	15.97	19.46	**7.89**	12.90	13.94	**21.49**	**14.05**	19.74
15_Chen	30.79	15.84	18.88	9.26	15.82	20.79	29.55	16.27	**18.51**
15_RW3DAS1	42.33	25.82	23.31	10.03	13.05	17.53	27.53	21.20	28.34
15_Adamiak	23.85	18.00	24.57	8.47	15.42	17.75	27.29	15.23	24.71
15_SimRNAAS1	51.04	23.12	27.81	12.48	11.61	18.40	34.12	30.04	29.17
15_SimRNAAS2	45.00	24.87	31.28	10.96	13.47	22.10	36.72	27.40	33.25
15_RW3DAS2	36.70	26.80	28.95	10.57	13.74	20.31	28.91	25.14	31.50
15_RNAComposerAS2	33.55	18.05	26.23	11.78	17.45	17.01	33.29	23.98	25.96
15_3dRNAAS2	57.32	29.09	56.08	27.89	24.26	38.73	49.05	32.37	40.11
rp18									
TorRNA	**28.72**	**16.06**	25.11	14.42	**12.15**	**11.83**	**15.29**	20.75	22.76
18_3dRNA	58.13	28.02	41.99	27.54	20.08	33.63	29.53	32.87	29.45
18_Dokholyan	58.70	20.70	56.70	19.49	17.06	17.01	26.33	32.46	35.61
18_LeeASmodel	32.74	20.65	**22.23**	15.52	22.50	19.74	24.77	27.13	30.49
18_Chen	45.62	23.46	29.48	12.41	15.48	34.77	37.96	21.22	22.79
18_SimRNA	42.58	25.09	29.84	14.26	19.69	20.72	31.25	26.77	30.57
18_Lee	32.74	20.65	**22.23**	15.52	22.50	19.74	24.77	27.13	30.49
18_RW3D	34.46	21.39	25.49	12.76	13.45	13.88	17.69	22.96	22.27
18_RNAComposer	35.11	23.00	31.00	14.01	17.47	21.78	31.24	26.24	33.13
18_YagoubAli	59.66	25.32	41.62	13.64	17.01	26.69	33.67	29.17	37.34
18_Das	34.83	23.32	25.47	**7.50**	12.61	14.66	18.08	**13.65**	**12.79**
18_Ding	42.59	22.21	33.80	17.38	18.65	15.64	21.13	28.01	28.68
rp07									
TorRNA	**38.87**	**17.76**	**28.41**	11.69	**14.41**	**13.62**	**23.66**	21.46	**25.11**
7_Adamiak	56.06	31.90	43.78	15.18	20.46	20.70	38.97	27.76	37.74
7_Ding	49.32	20.82	44.12	13.80	21.24	21.57	36.47	29.91	36.74
7_Dokholyan	59.24	21.73	55.14	17.86	20.25	18.49	33.31	35.54	35.88
7_Chen	40.27	23.02	31.37	**9.70**	15.49	21.29	32.23	**18.60**	30.01
rp11									
TorRNA	**32.31**	**15.88**	**29.29**	8.71	**12.79**	**10.83**	**15.63**	**13.97**	23.56
11_Xiao	56.07	24.57	39.79	11.95	14.26	28.47	31.87	24.94	28.56
11_Ding	48.16	18.68	45.85	12.53	15.99	21.44	25.54	24.17	33.19
11_Adamiak	54.68	30.76	47.30	**7.51**	19.18	25.06	29.17	24.16	30.47
11_Das	36.17	21.53	32.38	7.73	16.29	15.45	23.47	14.45	**21.74**
11_Chen	38.62	21.74	30.69	12.21	15.61	26.63	35.87	28.12	28.98
11_Bujnicki	35.26	18.47	37.21	14.92	16.19	18.44	24.22	18.13	33.06
rp04									
TorRNA	41.14	20.47	33.41	6.97	15.23	23.42	22.84	19.52	22.05
4_santalucia	41.44	26.44	31.04	6.70	19.27	26.54	26.58	9.08	10.96
4_adamiak	45.71	32.24	36.41	6.65	19.12	24.92	26.77	12.33	17.01
4_major	55.23	32.11	36.95	11.92	17.72	30.68	26.94	15.30	15.30
4_das	**34.02**	21.59	25.63	**6.22**	13.74	**19.14**	20.62	**7.05**	**8.95**
4_mikolajczak	51.90	27.88	40.37	7.22	19.84	29.52	35.05	18.52	22.90
4_bujnicki	36.84	22.39	29.26	6.35	16.63	25.32	25.16	10.01	14.90
4_dokholyan	38.80	19.97	31.64	6.53	15.47	20.03	21.28	11.94	14.34
4_chen	34.59	**18.60**	**25.49**	6.47	**12.35**	20.39	**19.87**	7.52	9.52
rp09									
TorRNA	**34.52**	**13.74**	**29.98**	18.50	17.88	16.71	35.39	27.33	35.91
9_Bujnicki	48.74	23.07	40.30	24.92	23.29	26.10	33.81	33.70	34.71
9_Das	44.02	20.05	32.80	**11.08**	**15.85**	**16.34**	**21.27**	**18.15**	**25.49**
9_Dokholyan	62.25	18.21	61.37	19.72	20.89	19.22	37.42	35.69	48.96
9_Chen	43.37	18.47	33.57	14.51	20.37	18.57	23.62	28.39	30.12
9_Ding	51.52	21.42	44.02	17.74	19.95	22.00	37.42	35.51	37.28
rp08									
TorRNA	**31.66**	**12.79**	**23.25**	**6.57**	**10.73**	**14.09**	27.66	19.83	23.64
8_Bujnicki	45.12	21.01	35.56	13.05	17.51	24.51	29.55	21.26	26.02
8_Das	40.19	20.56	32.50	6.65	12.08	20.57	**25.98**	**14.41**	**21.49**
8_Adamiak	48.42	24.83	37.34	9.47	18.05	24.29	34.31	19.72	30.67
8_Ding	49.11	20.79	38.26	11.82	14.75	19.63	38.18	32.91	34.27
8_Chen	48.50	22.54	37.30	9.94	15.48	25.51	33.81	23.39	26.84
8_Dokholyan	66.87	22.16	59.96	10.56	17.25	19.64	41.87	29.14	44.06

Bibliography

[1] Bettye L Smith. The importance of molecular structure and conformation: learning with scanning probe microscopy. *Progress in Biophysics and Molecular Biology*, 74(1):93–113, 2000. ISSN 0079-6107. doi: https://doi.org/10.1016/S0079-6107(00)00016-X. URL https://www.sciencedirect.com/science/article/pii/S007961070000016X. Single Molecule Biochemistry and Molecular Biology.

[2] Syed Sauban Ghani. A comprehensive review of database resources in chemistry. *Eclética Química*, 45(3):57–68, Jul. 2020. doi: 10.26850/1678-4618eqj.v45.3.2020.p57-68. URL https://revista.iq.unesp.br/ojs/index.php/ecletica/article/view/1124.

[3] Weerapong Phadungsukanan, Markus Kraft, Joe A. Townsend, and Peter Murray-Rust. The semantics of chemical markup language (cml) for computational chemistry : Compchem. *Journal of Cheminformatics*, 4(1):15, Aug 2012. ISSN 1758-2946. doi: 10.1186/1758-2946-4-15. URL https://doi.org/10.1186/1758-2946-4-15.

[4] M. Álvarez-Moreno, C. de Graaf, N. López, F. Maseras, J. M. Poblet, and C. Bo. Managing the computational chemistry big data problem: The iochem-bd platform. *Journal of Chemical Information and Modeling*, 55(1):95–103, Jan 2015. ISSN 1549-9596. doi: 10.1021/ci500593j. URL https://doi.org/10.1021/ci500593j.

[5] Carles Bo, Feliu Maseras, and Núria López. The role of computational results databases in accelerating the discovery of catalysts. *Nature Catalysis*, 1(11):809–810, Nov 2018. ISSN 2520-1158. doi: 10.1038/s41929-018-0176-4. URL https://doi.org/10.1038/s41929-018-0176-4.

[6] Rovshan G. Sadygov, Daniel Cociorva, and John R. Yates. Large-scale database searching using tandem mass spectra: Looking up the answer in the back of the book. *Nature Methods*, 1(3):195–202, Dec 2004. ISSN 1548-7105. doi: 10.1038/nmeth725. URL https://doi.org/10.1038/nmeth725.

[7] Matevz Pesek, Andraz Juvan, Jure Jakos, Janez Kosmrlj, Matija Marolt, and Martin Gazvoda. Database independent automated structure elucidation of organic molecules based on ir, 1h nmr,

13c nmr, and ms data. *Journal of chemical information and modeling*, 61(2):756–763, 2020. URL https://doi.org/10.1021/acs.jcim.0c01332.

[8] Michael A. Stravs, Kai Dührkop, Sebastian Böcker, and Nicola Zamboni. Msnovelist: de novo structure generation from mass spectra. *Nature Methods*, 19(7):865–870, Jul 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01486-3. URL https://doi.org/10.1038/s41592-022-01486-3.

[9] Sriram Devata, Bhuvanesh Sridharan, Sarvesh Mehta, Yashaswi Pathak, Siddhartha Laghuvarapu, Girish Varma, and U. Deva Priyakumar. Deepspinn - deep reinforcement learning for molecular structure prediction from infrared and 13c nmr spectra. *Digital Discovery*, pages –, 2024. doi: 10.1039/D4DD00008K. URL http://dx.doi.org/10.1039/D4DD00008K.

[10] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. doi: 10.1021/ci300415d. URL https://doi.org/10.1021/ci300415d. PMID: 23088335.

[11] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, Aug 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.22. URL https://doi.org/10.1038/sdata.2014.22.

[12] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, Jan 2020. ISSN 1476-4687. doi: 10.1038/s41586-019-1923-7. URL https://doi.org/10.1038/s41586-019-1923-7.

[13] Georgios Joannis Pappas and Shankar Subramaniam. Analysis of the effects of multiple sequence alignments in protein secondary structure prediction. In João Carlos Setubal and Sergio Verjovski-Almeida, editors, *Advances in Bioinformatics and Computational Biology*, pages 128–140, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31861-3. URL https://doi.org/10.1007/11532323_14.

[14] Xiaomin Fang, Fan Wang, Lihang Liu, Jingzhou He, Dayong Lin, Yingfei Xiang, Kunrui Zhu, Xiaonan Zhang, Hua Wu, Hui Li, and Le Song. A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. *Nature Machine Intelligence*, 5(10):1087–1096, Oct 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00721-6. URL https://doi.org/10.1038/s42256-023-00721-6.

[15] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, May 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

[16] Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, Irwin King, and Yu Li. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions, 2022. URL https://doi.org/10.48550/arXiv.2204.00300.

[17] Tao Shen, Zhihang Hu, Zhangzhi Peng, Jiayang Chen, Peng Xiong, Liang Hong, Liangzhen Zheng, Yixuan Wang, Irwin King, Sheng Wang, Siqi Sun, and Yu Li. E2efold-3d: End-to-end deep learning method for accurate de novo rna 3d structure prediction, 2022. URL https://doi.org/10.48550/arXiv.2207.01586.

[18] de Paula J Atkins PW. *Elements of physical chemistry (5th ed.)*. Oxford: Oxford U.P, 2009.

[19] Smith Janice Janice Gorzynski Smith. *Organic Chemistry*. McGraw-Hill.

[20] Brian F. Taylor Brian E. Mann. *13C NMR data for organometallic compounds*. Academic Press, 1981.

[21] *The Theory of NMR - Chemical Shift*. University of Colorado, Boulder, Chemistry and Biochemistry Department, 2011.

[22] Zhaorui Huang, Michael S. Chen, Cristian P. Woroch, Thomas E. Markland, and Matthew W. Kanan. A framework for automated structure elucidation from routine nmr spectra. *Chem. Sci.*, 12:15329–15338, 2021. doi: 10.1039/D1SC04105C. URL http://dx.doi.org/10.1039/D1SC04105C.

[23] Mikhail Elyashberg and Dimitris Argyropoulos. Computer assisted structure elucidation (case): Current and future perspectives. *Magnetic Resonance in Chemistry*, 59(7):669–690, 2021. doi: https://doi.org/10.1002/mrc.5115. URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/mrc.5115.

[24] Darcy C. Burns, Eugene P. Mazzola, and William F. Reynolds. The role of computer-assisted structure elucidation (case) programs in the structure elucidation of complex natural products. *Nat. Prod. Rep.*, 36:919–933, 2019. doi: 10.1039/C9NP00007K. URL http://dx.doi.org/10.1039/C9NP00007K.

[25] Christoph Steinbeck. Recent developments in automated structure elucidation of natural products. *Nat. Prod. Rep.*, 21:512–518, 2004. doi: 10.1039/B400678J. URL http://dx.doi.org/10.1039/B400678J.

[26] Youngchun Kwon, Dongseon Lee, Youn-Suk Choi, and Seokho Kang. Molecular search by nmr spectrum based on evaluation of matching between spectrum and molecule. *Scientific Reports*, 11(1):1–9, 2021. URL https://doi.org/10.1038/s41598-021-00488-z.

[27] Kai Dührkop, Huibin Shen, Marvin Meusel, Juho Rousu, and Sebastian Böcker. Searching molecular structure databases with tandem mass spectra using csi: Fingerid. *Proceedings of the National Academy of Sciences*, 112(41):12580–12585, 2015. URL https://doi.org/10.1073/pnas.1509788112.

[28] Mikhail E Elyashberg, Antony Williams, and Kirill Blinov. *Contemporary Computer-Assisted Approaches to Molecular Structure Elucidation*. New Developments in NMR. The Royal Society of Chemistry, 2012. ISBN 978-1-84973-432-5. doi: 10.1039/9781849734578. URL http://dx.doi.org/10.1039/9781849734578.

[29] Markus C. Hemmer and Johann Gasteiger. Prediction of three-dimensional molecular structures using information from infrared spectra. *Analytica Chimica Acta*, 420(2):145–154, 2000. ISSN 0003-2670. doi: https://doi.org/10.1016/S0003-2670(00)00876-X. URL https://www.sciencedirect.com/science/article/pii/S000326700000876X.

[30] Marilia Valli, Helena Mannochio Russo, Alan Cesar Pilon, Meri Emili Ferreira Pinto, Nathalia B. Dias, Rafael Teixeira Freire, Ian Castro-Gamboa, and Vanderlan da Silva Bolzani. Computational methods for nmr and ms for structure elucidation i: software for basic nmr. *Physical Sciences Reviews*, 4(10):20180108, 2019. doi: doi:10.1515/psr-2018-0108. URL https://doi.org/10.1515/psr-2018-0108.

[31] Marilia Valli, Helena Mannochio Russo, Alan Cesar Pilon, Meri Emili Ferreira Pinto, Nathalia B. Dias, Rafael Teixeira Freire, Ian Castro-Gamboa, and Vanderlan da Silva Bolzani. Computational methods for nmr and ms for structure elucidation ii: database resources and advanced methods. *Physical Sciences Reviews*, 4(11):20180167, 2019. doi: doi:10.1515/psr-2018-0167. URL https://doi.org/10.1515/psr-2018-0167.

[32] Gabin T. M. Bitchagno and Serge Alain Fobofou Tanemossu. Computational methods for nmr and ms for structure elucidation iii: More advanced approaches. *Physical Sciences Reviews*, 4

(9):20180109, 2019. doi: doi:10.1515/psr-2018-0109. URL https://doi.org/10.1515/psr-2018-0109.

[33] M.E Elyashberg, K.A Blinov, and E.R Martirosian. A new approach to computer-aided molecular structure elucidation: the expert system structure elucidator. *Laboratory Automation & Information Management*, 34(1):15–30, 1999. ISSN 1381-141X. doi: https://doi.org/10.1016/S1381-141X(99)00002-7. URL https://www.sciencedirect.com/science/article/pii/S1381141X99000027.

[34] Lorena Martins Guimarães Moreira and Jochen Junker. Sampling case application for the quality control of published natural product structures. *Molecules*, 26(24):7543, 2021. URL https://doi.org/10.3390/molecules26247543.

[35] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, and Shanrong Zhao. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, Jun 2019. ISSN 1474-1784. doi: 10.1038/s41573-019-0024-5. URL https://doi.org/10.1038/s41573-019-0024-5.

[36] Sean Ekins, Ana C. Puhl, Kimberley M. Zorn, Thomas R. Lane, Daniel P. Russo, Jennifer J. Klein, Anthony J. Hickey, and Alex M. Clark. Exploiting machine learning for end-to-end drug discovery and development. *Nature Materials*, 18(5):435–441, May 2019. ISSN 1476-4660. doi: 10.1038/s41563-019-0338-z. URL https://doi.org/10.1038/s41563-019-0338-z.

[37] Sarvesh Mehta, Siddhartha Laghuvarapu, Yashaswi Pathak, Aaftaab Sethi, Mallika Alvala, and U. Deva Priyakumar. Memes: Machine learning framework for enhanced molecular screening. *Chem. Sci.*, 12:11710–11721, 2021. doi: 10.1039/D1SC02783B. URL http://dx.doi.org/10.1039/D1SC02783B.

[38] Sergei Manzhos and Tucker Carrington. Neural network potential energy surfaces for small molecules and reactions. *Chemical Reviews*, 121(16):10187–10217, Aug 2021. ISSN 0009-2665. doi: 10.1021/acs.chemrev.0c00665. URL https://doi.org/10.1021/acs.chemrev.0c00665.

[39] Punyaslok Pattnaik, Shampa Raghunathan, Tarun Kalluri, Prabhakar Bhimalapuram, C. V. Jawahar, and U. Deva Priyakumar. Machine learning for accurate force calculations in molecular dynamics simulations. *The Journal of Physical Chemistry A*, 124(34):6954–6967, Aug 2020. ISSN 1089-5639. doi: 10.1021/acs.jpca.0c03926. URL https://doi.org/10.1021/acs.jpca.0c03926.

[40] Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual Review of Physical Chemistry*, 71(1):361–390, 2020.

doi: 10.1146/annurev-physchem-042018-052331. URL https://doi.org/10.1146/annurev-physchem-042018-052331. PMID: 32092281.

[41] Yashas B. L. Samaga, Shampa Raghunathan, and U. Deva Priyakumar. Scones: Self-consistent neural network for protein stability prediction upon mutation. *The Journal of Physical Chemistry B*, 125(38):10657–10671, Sep 2021. ISSN 1520-6106. doi: 10.1021/acs.jpcb.1c04913. URL https://doi.org/10.1021/acs.jpcb.1c04913.

[42] Rishal Aggarwal, Akash Gupta, Vineeth Chelur, C. V. Jawahar, and U. Deva Priyakumar. Deeppocket: Ligand binding site detection and segmentation using 3d convolutional neural networks. *Journal of Chemical Information and Modeling*, Aug 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00799. URL https://doi.org/10.1021/acs.jcim.1c00799.

[43] Yashaswi Pathak, Siddhartha Laghuvarapu, Sarvesh Mehta, and U. Deva Priyakumar. Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):873–880, Apr. 2020. doi: 10.1609/aaai.v34i01.5433. URL https://ojs.aaai.org/index.php/AAAI/article/view/5433.

[44] Siddhartha Laghuvarapu, Yashaswi Pathak, and U. Deva Priyakumar. Band nn: A deep learning framework for energy prediction and geometry optimization of organic small molecules. *Journal of Computational Chemistry*, 41(8):790–799, 2020. doi: https://doi.org/10.1002/jcc.26128. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.26128.

[45] Manan Goel, Shampa Raghunathan, Siddhartha Laghuvarapu, and U. Deva Priyakumar. Molegular: Molecule generation using reinforcement learning with alternating rewards. *Journal of Chemical Information and Modeling*, 61(12):5815–5826, Dec 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c01341. URL https://doi.org/10.1021/acs.jcim.1c01341.

[46] D. Ricard, C. Cachet, D. Cabrol-Bass, and T. P. Forrest. Neural network approach to structural feature recognition from infrared spectra. *Journal of Chemical Information and Computer Sciences*, 33(2):202–210, Mar 1993. ISSN 0095-2338. doi: 10.1021/ci00012a004. URL https://doi.org/10.1021/ci00012a004.

[47] Hao Ren, Hao Li, Qian Zhang, Lijun Liang, Wenyue Guo, Fang Huang, Yi Luo, and Jun Jiang. A machine learning vibrational spectroscopy protocol for spectrum prediction and spectrum-based structure recognition. *Fundamental Research*, 1(4):488–494, 2021. ISSN 2667-3258. doi: https://doi.org/10.1016/j.fmre.2021.05.005. URL https://www.sciencedirect.com/science/article/pii/S2667325821000972.

[48] Kun Yao, John E. Herr, David W. Toth, Ryker Mckintyre, and John Parkhill. The tensormol-0.1 model chemistry: a neural network augmented with long-range physics. *Chem. Sci.*, 9:

2261–2269, 2018. doi: 10.1039/C7SC04934J. URL http://dx.doi.org/10.1039/C7SC04934J.

[49] Alexei A. Kananenka, Kun Yao, Steven A. Corcelli, and J. L. Skinner. Machine learning for vibrational spectroscopic maps. *Journal of Chemical Theory and Computation*, 15(12):6850–6858, Dec 2019. ISSN 1549-9618. doi: 10.1021/acs.jctc.9b00698. URL https://doi.org/10.1021/acs.jctc.9b00698.

[50] Michael Gastegger, Jörg Behler, and Philipp Marquetand. Machine learning molecular dynamics for the simulation of infrared spectra. *Chem. Sci.*, 8:6924–6935, 2017. doi: 10.1039/C7SC02267K. URL http://dx.doi.org/10.1039/C7SC02267K.

[51] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller. Schnetpack: A deep learning toolbox for atomistic systems. *Journal of Chemical Theory and Computation*, 15(1):448–455, Jan 2019. ISSN 1549-9618. doi: 10.1021/acs.jctc.8b00908. URL https://doi.org/10.1021/acs.jctc.8b00908.

[52] Paruzzo F.M., Hofstetter A., Musil F., De S., Ceriotti M., and Emsley L. Chemical shifts in molecular solids by machine learning. *Nature Communications*, 9(1), 2018. doi: 10.1038/s41467-018-06972-x. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85055612214&doi=10.1038%2fs41467-018-06972-x&partnerID=40&md5=ddbbb3defe5956c55d18e4ac2ae69e25. All Open Access, Gold Open Access, Green Open Access.

[53] Eric Jonas and Stefan Kuhn. Rapid prediction of nmr spectral properties with quantified uncertainty. *Journal of Cheminformatics*, 11(1):50, Aug 2019. ISSN 1758-2946. doi: 10.1186/s13321-019-0374-3. URL https://doi.org/10.1186/s13321-019-0374-3.

[54] Ziyue Yang, Maghesree Chakraborty, and Andrew D. White. Predicting chemical shifts with graph neural networks. *Chem. Sci.*, 12:10802–10809, 2021. doi: 10.1039/D1SC01895G. URL http://dx.doi.org/10.1039/D1SC01895G.

[55] Sheng Ye, Wei Hu, Xin Li, Jinxiao Zhang, Kai Zhong, Guozhen Zhang, Yi Luo, Shaul Mukamel, and Jun Jiang. A neural network protocol for electronic excitations of ¡i¿n¡/i¿-methylacetamide. *Proceedings of the National Academy of Sciences*, 116(24):11612–11617, 2019. doi: 10.1073/pnas.1821044116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1821044116.

[56] Kunal Ghosh, Annika Stuke, Milica Todorović, Peter Bjørn Jørgensen, Mikkel N. Schmidt, Aki Vehtari, and Patrick Rinke. Deep learning spectroscopy: Neural networks for molecular excitation spectra. *Advanced Science*, 6(9):1801367, 2019. doi: https://doi.org/10.1002/advs.201801367. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.201801367.

[57] Kumar Giri S., Saalmann U., and Rost J.M. Purifying electron spectra from noisy pulses with machine learning using synthetic hamilton matrices. *Physical Review Letters*, 124(11), 2020. doi: 10.1103/PhysRevLett.124.113201. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083041528&doi=10.1103%2fPhysRevLett.124.113201&partnerID=40&md5=dceddc75dfa28758501a3bdfa3282459. Cited by: 7; All Open Access, Green Open Access, Hybrid Gold Open Access.

[58] Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging, 2020. URL https://arxiv.org/abs/2005.06001.

[59] Zhimeng Wang, Xiaoyu Feng, Junhong Liu, Minchun Lu, and Menglong Li. Functional groups prediction from infrared spectra based on computer-assist approaches. *Microchemical Journal*, 159:105395, 2020. ISSN 0026-265X. doi: https://doi.org/10.1016/j.microc.2020.105395. URL https://www.sciencedirect.com/science/article/pii/S0026265X20320853.

[60] Jonathan A. Fine, Anand A. Rajasekar, Krupal P. Jethava, and Gaurav Chopra. Spectral deep learning for prediction and prospective validation of functional groups. *Chem. Sci.*, 11:4618–4630, 2020. doi: 10.1039/C9SC06240H. URL http://dx.doi.org/10.1039/C9SC06240H.

[61] Eric Jonas. Deep imitation learning for molecular inverse problems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/b0bef4c9a6e50d43880191492d4fc827-Paper.pdf.

[62] Alexander Howarth, Kristaps Ermanis, and Jonathan M. Goodman. Dp4-ai automated nmr data analysis: straight from spectrometer to structure. *Chem. Sci.*, 11:4351–4359, 2020. doi: 10.1039/D0SC00442A. URL http://dx.doi.org/10.1039/D0SC00442A.

[63] Jinzhe Zhang, Kei Terayama, Masato Sumita, Kazuki Yoshizoe, Kengo Ito, Jun Kikuchi, and Koji Tsuda. Nmr-ts: de novo molecule identification from nmr spectra. *Science and Technology of Advanced Materials*, 21(1):552–561, 2020. doi: 10.1080/14686996.2020.1793382. URL https://doi.org/10.1080/14686996.2020.1793382.

[64] Xiufeng Yang, Jinzhe Zhang, Kazuki Yoshizoe, Kei Terayama, and Koji Tsuda. Chemts: an efficient python library for de novo molecular generation. *Science and technology of advanced materials*, 18(1):972–976, 2017. URL https://doi.org/10.1080%2F14686996.2017.1401424.

[65] Mikhail Elyashberg, Kirill Blinov, Sergey Molodtsov, Yegor Smurnyy, Antony J. Williams, and Tatiana Churanova. Computer-assisted methods for molecular structure elucidation: realizing a spectroscopist's dream. *Journal of Cheminformatics*, 1(1):3, Mar 2009. ISSN 1758-2946. doi: 10.1186/1758-2946-1-3. URL https://doi.org/10.1186/1758-2946-1-3.

[66] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017. URL https://arxiv.org/abs/1712.01815.

[67] Marvin Alberts, Teodoro Laino, and Alain C. Vaucher. Leveraging infrared spectroscopy for automated structure elucidation, 2023. URL https://doi.org/10.26434/chemrxiv-2023-5v27f.

[68] Marvin Alberts, Federico Zipoli, and Alain C. Vaucher. Learning the language of nmr: Structure elucidation from nmr spectra using transformer models, 2023. URL https://doi.org/10.26434/chemrxiv-2023-5v27f.

[69] Bhuvanesh Sridharan, Sarvesh Mehta, Yashaswi Pathak, and U. Deva Priyakumar. Deep reinforcement learning for molecular inverse problem of nuclear magnetic resonance spectra to molecular structure. *The Journal of Physical Chemistry Letters*, pages 4924–4933, May 2022. doi: 10.1021/acs.jpclett.2c00624. URL https://doi.org/10.1021/acs.jpclett.2c00624.

[70] Stefan Kuhn and Nils E. Schlörer. 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*, 53(8):582–589, 2015. doi: https://doi.org/10.1002/mrc.4263. URL https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/mrc.4263.

[71] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017. URL http://arxiv.org/abs/1704.01212.

[72] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, Nov 2012. ISSN 1549-9596. doi: 10.1021/ci300415d. URL https://doi.org/10.1021/ci300415d.

[73] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L.

Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox. Gaussian 09, Revision A.1, 2016. Gaussian Inc. Wallingford CT.

[74] Charles McGill, Michael Forsuelo, Yanfei Guan, and William H. Green. Predicting infrared spectra with message passing neural networks. *Journal of Chemical Information and Modeling*, 61(6):2594–2609, Jun 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00055. URL https://doi.org/10.1021/acs.jcim.1c00055.

[75] Chu P.M., Guenther F.R., Rhoderick G.C., and Lafferty W.J. The nist quantitative infrared database. *Journal of Research of the National Institute of Standards and Technology*, 104(1): 59–81, 1999. URL https://doi.org/10.6028/jres.104.004.

[76] Sonjae Wallace, Samuel G. Lambrakos, Andrew Shabaev, and Lou Massa. On using dft to construct an ir spectrum database for pfas molecules. *Structural Chemistry*, 33(1):247–256, Feb 2022. ISSN 1572-9001. doi: 10.1007/s11224-021-01844-5. URL https://doi.org/10.1007/s11224-021-01844-5.

[77] Amit Gupta, Sabyasachi Chakraborty, and Raghunathan Ramakrishnan. 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*, 2(3): 035010, may 2021. doi: 10.1088/2632-2153/abe347. URL https://doi.org/10.1088/2632-2153/abe347.

[78] Michael Mehring. *High resolution NMR spectroscopy in solids*, volume 11. Springer Science & Business Media, 2012.

[79] Herman Rull, Markus Fischer, and Stefan Kuhn. Nmr shift prediction from small data quantities. *Journal of Cheminformatics*, 15(1):114, Nov 2023. ISSN 1758-2946. doi: 10.1186/s13321-023-00785-x. URL https://doi.org/10.1186/s13321-023-00785-x.

[80] Yanfei Guan, S. V. Shree Sowndarya, Liliana C. Gallegos, Peter C. St. John, and Robert S. Paton. Real-time prediction of 1h and 13c chemical shifts with dft accuracy using a 3d graph neural network. *Chem. Sci.*, 12:12012–12026, 2021. doi: 10.1039/D1SC03343C. URL http://dx.doi.org/10.1039/D1SC03343C.

[81] Michael W Lodewyk, Matthew R Siebert, and Dean J Tantillo. Computational prediction of 1H and 13C chemical shifts: a useful tool for natural product, mechanistic, and synthetic organic

chemistry. *Chem Rev*, 112(3):1839–1862, November 2011. URL https://doi.org/10.1021/cr200106v.

[82] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan 2016. URL https://doi.org/10.1038/nature16961.

[83] Steven James, George Konidaris, and Benjamin Rosman. An analysis of monte carlo tree search. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3576–3582. AAAI Press, 2017. URL https://doi.org/10.1609/aaai.v31i1.11028.

[84] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

[85] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, Andrew Palmer, Volker Settels, Tommi Jaakkola, Klavs Jensen, and Regina Barzilay. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, Aug 2019. ISSN 1549-9596. doi: 10.1021/acs.jcim.9b00237. URL https://doi.org/10.1021/acs.jcim.9b00237.

[86] Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. *CoRR*, abs/1603.05629, 2016. URL http://arxiv.org/abs/1603.05629.

[87] Chein-I Chang. An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis. *IEEE Transactions on Information Theory*, 46(5):1927–1932, 2000. doi: 10.1109/18.857802. URL https://doi.org/10.1109/18.857802.

[88] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 282–293, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-46056-5. URL https://doi.org/10.1007/11871842_29.

[89] Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, Mar 2018. ISSN 1476-4687. doi: 10.1038/nature25978. URL https://doi.org/10.1038/nature25978.

[90] Thomas M Moerland, Joost Broekens, Aske Plaat, and Catholijn M Jonker. Monte Carlo Tree Search for Asymmetric Trees. *arXiv preprint arXiv:1805.09218*, 2018. URL https://doi.org/10.48550/arXiv.1805.09218.

[91] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, Oct 2017. ISSN 1476-4687. doi: 10.1038/nature24270. URL https://doi.org/10.1038/nature24270.

[92] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/f9be311e65d81a9ad8150a60844bb94c-Paper.pdf.

[93] Greg Landrum. Rdkit: Open-source cheminformatics software. 2016. URL https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

[94] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://doi.org/10.48550/arXiv.1412.6980.

[95] Steven H Bertz. The first general index of molecular complexity. *Journal of the American Chemical Society*, 103(12):3599–3601, 1981. URL https://doi.org/10.1021/ja00402a071.

[96] Nils G Walter and David R Engelke. Ribozymes: catalytic RNAs that cut things, make things, and do odd and useful jobs. *Biologist (London)*, 49(5):199–203, October 2002. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3770912/.

[97] Yicheng Long, Xueyin Wang, Daniel T Youmans, and Thomas R Cech. How do lncrnas regulate transcription? *Science advances*, 3(9):eaao2110, 2017. URL https://doi.org/10.1126%2Fsciadv.aao2110.

[98] Yi Shu, Fengmei Pi, Ashwani Sharma, Mehdi Rajabi, Farzin Haque, Dan Shu, Markos Leggas, B. Mark Evers, and Peixuan Guo. Stable rna nanoparticles as potential new generation drugs for cancer therapy. *Advanced Drug Delivery Reviews*, 66:74–89, 2014. ISSN 0169-409X. doi: https://doi.org/10.1016/j.addr.2013.11.006. URL https://www.sciencedirect.com/science/article/pii/S0169409X13002652. Cancer nanotechnology.

[99] Muskan Muskan, Pevindu Abeysinghe, Riccardo Cecchin, Heather Branscome, Kevin V Morris, and Fatah Kashanchi. Therapeutic potential of rna-enriched extracellular vesicles: The next generation in rna delivery via biogenic nanoparticles. *Molecular Therapy*, 2024. URL https://doi.org/10.1016/j.ymthe.2024.02.025.

[100] Luca Mollica, Francesca Anna Cupaioli, Grazisa Rossetti, and Federica Chiappori. An overview of structural approaches to study therapeutic rnas. *Frontiers in Molecular Biosciences*, 9, 2022. ISSN 2296-889X. doi: 10.3389/fmolb.2022.1044126. URL https://www.frontiersin.org/articles/10.3389/fmolb.2022.1044126.

[101] Hannah Zogg, Rajan Singh, and Seungil Ro. Current advances in rna therapeutics for human diseases. *International Journal of Molecular Sciences*, 23(5), 2022. ISSN 1422-0067. doi: 10.3390/ijms23052736. URL https://www.mdpi.com/1422-0067/23/5/2736.

[102] Jessica L. Childs-Disney, Xueyi Yang, Quentin M. R. Gibaut, Yuquan Tong, Robert T. Batey, and Matthew D. Disney. Targeting rna structures with small molecules. *Nature Reviews Drug Discovery*, 21(10):736–762, Oct 2022. ISSN 1474-1784. doi: 10.1038/s41573-022-00521-4. URL https://doi.org/10.1038/s41573-022-00521-4.

[103] Leandro Grille, Diego Gallego, Leonardo Darré, Gabriela da Rosa, Federica Battistini, Modesto Orozco, and Pablo D. Dans. The pseudo-torsional space of rna. *bioRxiv*, 2022. doi: 10.1101/2022.06.24.497007. URL https://www.biorxiv.org/content/early/2022/06/28/2022.06.24.497007.

[104] Stefanie A Mortimer, Mary Anne Kidwell, and Jennifer A Doudna. Insights into rna structure and function from genome-wide studies. *Nature Reviews Genetics*, 15(7):469–479, 2014. URL https://doi.org/10.1038/nrg3681.

[105] Ligang Wu and Joel G Belasco. Let me count the ways: mechanisms of gene regulation by mirnas and sirnas. *Molecular cell*, 29(1):1–7, 2008. URL https://doi.org/10.1016/j.molcel.2007.12.010.

[106] Ignacio Tinoco and Carlos Bustamante. How rna folds. *Journal of Molecular Biology*, 293(2):271–281, 1999. ISSN 0022-2836. doi: https://doi.org/10.1006/jmbi.1999.3001. URL https://www.sciencedirect.com/science/article/pii/S0022283699930012.

[107] Jinsong Zhang, Yuhan Fei, Lei Sun, and Qiangfeng Cliff Zhang. Advances and opportunities in rna structure experimental determination and computational modeling. *Nature Methods*, 19(10):1193–1207, Oct 2022. ISSN 1548-7105. doi: 10.1038/s41592-022-01623-y. URL https://doi.org/10.1038/s41592-022-01623-y.

[108] Haiyun Ma, Xinyu Jia, Kaiming Zhang, and Zhaoming Su. Cryo-em advances in rna structure determination. *Signal Transduction and Targeted Therapy*, 7(1):58, 2022. URL https://doi.org/10.1038/s41392-022-00916-0.

[109] Ruth Nussinov, George Pieczenik, Jerrold R Griggs, and Daniel J Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied mathematics*, 35(1):68–82, 1978. URL https://www.jstor.org/stable/2101031.

[110] Michael Zuker and Patrick Stiegler. Optimal computer folding of large rna sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–148, 1981. URL https://doi.org/10.1093%2Fnar%2F9.1.133.

[111] Michael Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13):3406–3415, 2003. URL https://doi.org/10.1093/nar/gkg595.

[112] Nicholas R Markham and Michael Zuker. Unafold: software for nucleic acid folding and hybridization. *Bioinformatics: structure, function and applications*, pages 3–31, 2008. URL https://doi.org/10.1007/978-1-60327-429-6_1.

[113] Ivo L Hofacker. Vienna rna secondary structure server. *Nucleic acids research*, 31(13):3429–3431, 2003. URL https://doi.org/10.1093/nar/gkg599.

[114] Jessica S Reuter and David H Mathews. Rnastructure: software for rna secondary structure prediction and analysis. *BMC bioinformatics*, 11(1):1–9, 2010. URL https://doi.org/10.1186/1471-2105-11-129.

[115] Yoshiyasu Takefuji and L Chen. Parallel algorithms for finding a near-maximum independent set of a circle graph. *IEEE Trans. Neural Networks*, 1(3):263, 1990. URL https://doi.org/10.1109/72.80251.

[116] EW Steeg. Artificial intelligence and molecular biology. In *chapter Neural Networks, Adaptive Optimization, and RNA Secondary Structure Prediction*, pages 121–60. American Association for Artificial Intelligence, 1993.

[117] Tianbing Xia, John SantaLucia Jr, Mark E Burkard, Ryszard Kierzek, Susan J Schroeder, Xiaoqi Jiao, Christopher Cox, and Douglas H Turner. Thermodynamic parameters for an expanded nearest-neighbor model for formation of rna duplexes with watson- crick base pairs. *Biochemistry*, 37(42):14719–14735, 1998. URL https://doi.org/10.1021/bi9809425.

[118] Piotr Klukowski, Roland Riek, and Peter Güntert. Rapid protein assignments and structures from raw nmr spectra with the deep learning technique artina. *Nature Communications*, 13(1):6151, Oct 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-33879-5. URL https://doi.org/10.1038/s41467-022-33879-5.

[119] Jaswinder Singh, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Rna secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature Communications*, 10(1):5407, Nov 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-13395-9. URL https://doi.org/10.1038/s41467-019-13395-9.

[120] Xinshi Chen, Yu Li, Ramzan Umarov, Xin Gao, and Le Song. Rna secondary structure prediction by learning unrolled algorithms, 2020. URL https://doi.org/10.48550/arXiv.2002.05810.

[121] Kengo Sato, Manato Akiyama, and Yasubumi Sakakibara. Rna secondary structure prediction using deep learning with thermodynamic integration. *Nature Communications*, 12(1):941, Feb 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-21194-4. URL https://doi.org/10.1038/s41467-021-21194-4.

[122] Jaswinder Singh, Kuldip Paliwal, Jaspreet Singh, and Yaoqi Zhou. Rna backbone torsion and pseudotorsion angle prediction using dilated convolutional neural networks. *Journal of Chemical Information and Modeling*, 61(6):2610–2622, Jun 2021. ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00153. URL https://doi.org/10.1021/acs.jcim.1c00153.

[123] Leven M. Wadley, Kevin S. Keating, Carlos M. Duarte, and Anna Marie Pyle. Evaluating and learning from rna pseudotorsional space: Quantitative validation of a reduced representation for rna structure. *Journal of Molecular Biology*, 372(4):942–957, 2007. ISSN 0022-2836. doi: https://doi.org/10.1016/j.jmb.2007.06.058. URL https://www.sciencedirect.com/science/article/pii/S0022283607008509.

[124] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0. URL https://doi.org/10.48550/arXiv.1603.05027.

[125] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions, 2016. URL https://doi.org/10.48550/arXiv.1511.07122.

[126] Peter W. Rose, Andreas Prlić, Ali Altunkaya, Chunxiao Bi, Anthony R. Bradley, Cole H. Christie, Luigi Di Costanzo, Jose M. Duarte, Shuchismita Dutta, Zukang Feng, Rachel Kramer Green, David S. Goodsell, Brian Hudson, Tara Kalro, Robert Lowe, Ezra Peisach, Christopher Randle, Alexander S. Rose, Chenghua Shao, Yi-Ping Tao, Yana Valasatava, Maria Voigt, John D. Westbrook, Jesse Woo, Huangwang Yang, Jasmine Y. Young, Christine Zardecki, Helen M. Berman, and Stephen K. Burley. The RCSB protein data bank: integrative view of protein, gene and 3D structural information. *Nucleic Acids Research*, 45(D1):D271–D281, 10 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw1000. URL https://doi.org/10.1093/nar/gkw1000.

[127] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://doi.org/10.48550/arXiv.1706.03762.

[128] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

[129] Neocles B. Leontis and Craig L. Zirbel. *Nonredundant 3D Structure Datasets for RNA Knowledge Extraction and Benchmarking*, pages 281–298. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-25740-7. doi: 10.1007/978-3-642-25740-7_13. URL https://doi.org/10.1007/978-3-642-25740-7_13.

[130] Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 03 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp163. URL https://doi.org/10.1093/bioinformatics/btp163.

[131] Limin Fu, Beifang Niu, Zhengwei Zhu, Sitao Wu, and Weizhong Li. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, October 2012. URL https://doi.org/10.1093/bioinformatics/bts565.

[132] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990. ISSN 0022-2836. doi: https://doi.org/10.1016/S0022-2836(05)80360-2. URL https://www.sciencedirect.com/science/article/pii/S0022283605803602.

[133] Marcin Magnus, Maciej Antczak, Tomasz Zok, Jakub Wiedemann, Piotr Lukasiak, Yang Cao, Janusz M Bujnicki, Eric Westhof, Marta Szachniuk, and Zhichao Miao. RNA-Puzzles toolkit: a computational resource of RNA 3D structure benchmark datasets, structure manipulation, and evaluation tools. *Nucleic Acids Research*, 48(2):576–588, 12 2019. ISSN 0305-1048. doi: 10.1093/nar/gkz1108. URL https://doi.org/10.1093/nar/gkz1108.

[134] José Almeida Cruz, Marc-Frédérick Blanchet, Michal Boniecki, Janusz M Bujnicki, Shi-Jie Chen, Song Cao, Rhiju Das, Feng Ding, Nikolay V Dokholyan, Samuel Coulbourn Flores, Lili Huang, Christopher A Lavender, Véronique Lisi, François Major, Katarzyna Mikolajczak, Dinshaw J Patel, Anna Philips, Tomasz Puton, John Santalucia, Fredrick Sijenyi, Thomas Hermann, Kristian Rother, Magdalena Rother, Alexander Serganov, Marcin Skorupski, Tomasz Soltysinski, Parin Sripakdeevong, Irina Tuszynska, Kevin M Weeks, Christina Waldsich, Michael

Wildauer, Neocles B Leontis, and Eric Westhof. RNA-Puzzles: a CASP-like evaluation of RNA three-dimensional structure prediction. *RNA*, 18(4):610–625, April 2012. URL https://doi.org/10.1261%2Frna.031054.111.

[135] Zhichao Miao, Ryszard W Adamiak, Marc-Frédérick Blanchet, Michal Boniecki, Janusz M Bujnicki, Shi-Jie Chen, Clarence Cheng, Grzegorz Chojnowski, Fang-Chieh Chou, Pablo Cordero, José Almeida Cruz, Adrian R Ferré-D'Amaré, Rhiju Das, Feng Ding, Nikolay V Dokholyan, Stanislaw Dunin-Horkawicz, Wipapat Kladwang, Andrey Krokhotin, Grzegorz Lach, Marcin Magnus, François Major, Thomas H Mann, Benoît Masquida, Dorota Matelska, Mélanie Meyer, Alla Peselis, Mariusz Popenda, Katarzyna J Purzycka, Alexander Serganov, Juliusz Stasiewicz, Marta Szachniuk, Arpit Tandon, Siqi Tian, Jian Wang, Yi Xiao, Xiaojun Xu, Jinwei Zhang, Peinan Zhao, Tomasz Zok, and Eric Westhof. RNA-Puzzles round II: assessment of RNA structure prediction programs applied to three large RNA structures. *RNA*, 21(6):1066–1084, June 2015. URL https://doi.org/10.1261%2Frna.049502.114.

[136] Zhichao Miao, Ryszard W Adamiak, Maciej Antczak, Robert T Batey, Alexander J Becka, Marcin Biesiada, Michał J Boniecki, Janusz M Bujnicki, Shi-Jie Chen, Clarence Yu Cheng, Fang-Chieh Chou, Adrian R Ferré-D'Amaré, Rhiju Das, Wayne K Dawson, Feng Ding, Nikolay V Dokholyan, Stanisław Dunin-Horkawicz, Caleb Geniesse, Kalli Kappel, Wipapat Kladwang, Andrey Krokhotin, Grzegorz E Łach, François Major, Thomas H Mann, Marcin Magnus, Katarzyna Pachulska-Wieczorek, Dinshaw J Patel, Joseph A Piccirilli, Mariusz Popenda, Katarzyna J Purzycka, Aiming Ren, Greggory M Rice, John Santalucia, Jr, Joanna Sarzynska, Marta Szachniuk, Arpit Tandon, Jeremiah J Trausch, Siqi Tian, Jian Wang, Kevin M Weeks, Benfeard Williams, 2nd, Yi Xiao, Xiaojun Xu, Dong Zhang, Tomasz Zok, and Eric Westhof. RNA-Puzzles round III: 3D RNA structure prediction of five riboswitches and one ribozyme. *RNA*, 23(5):655–672, May 2017. URL https://doi.org/10.1261%2Frna.060368.116.

[137] Zhichao Miao, Ryszard W Adamiak, Maciej Antczak, Michał J Boniecki, Janusz Bujnicki, Shi-Jie Chen, Clarence Yu Cheng, Yi Cheng, Fang-Chieh Chou, Rhiju Das, Nikolay V Dokholyan, Feng Ding, Caleb Geniesse, Yangwei Jiang, Astha Joshi, Andrey Krokhotin, Marcin Magnus, Olivier Mailhot, Francois Major, Thomas H Mann, Pawel Pikatkowski, Radoslaw Pluta, Mariusz Popenda, Joanna Sarzynska, Lizhen Sun, Marta Szachniuk, Siqi Tian, Jian Wang, Jun Wang, Andrew M Watkins, Jakub Wiedemann, Yi Xiao, Xiaojun Xu, Joseph D Yesselman, Dong Zhang, Yi Zhang, Zhenzhen Zhang, Chenhan Zhao, Peinan Zhao, Yuanzhe Zhou, Tomasz Zok, Adriana Żyła, Aiming Ren, Robert T Batey, Barbara L Golden, Lin Huang, David M Lilley, Yijin Liu, Dinshaw J Patel, and Eric Westhof. RNA-Puzzles round IV: 3D structure predictions of four ribozymes and two aptamers. *RNA*, 26(8):982–995, August 2020. URL https://doi.org/10.1261%2Frna.075341.120.

[138] Xiang-Jun Lu and Wilma K. Olson. 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Research*, 31(17):5108–

5121, 09 2003. ISSN 0305-1048. doi: 10.1093/nar/gkg680. URL https://doi.org/10.1093/nar/gkg680.

[139] Xiang-Jun Lu and Wilma K. Olson. 3dna: a versatile, integrated software system for the analysis, rebuilding and visualization of three-dimensional nucleic-acid structures. *Nature Protocols*, 3(7): 1213–1227, Jul 2008. ISSN 1750-2799. doi: 10.1038/nprot.2008.104. URL https://doi.org/10.1038/nprot.2008.104.

[140] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://doi.org/10.48550/arXiv.1810.04805.

[141] Yong-Chang Xu, Tian-Jun ShangGuan, Xue-Ming Ding, and Ngaam J. Cheung. Accurate prediction of protein torsion angles using evolutionary signatures and recurrent neural network. *Scientific Reports*, 11(1):21033, Oct 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-00477-2. URL https://doi.org/10.1038/s41598-021-00477-2.

[142] Emidio Capriotti, Tomas Norambuena, Marc A. Marti-Renom, and Francisco Melo. All-atom knowledge-based potential for RNA structure prediction and assessment. *Bioinformatics*, 27(8): 1086–1093, 02 2011. ISSN 1367-4803. doi: 10.1093/bioinformatics/btr093. URL https://doi.org/10.1093/bioinformatics/btr093.

[143] A Sali and TL Blundell. Comparative protein modelling by satisfaction of spatial restraints.(1993). *J. Mol. Biol*, 234:779. URL https://doi.org/10.1006/jmbi.1993.1626.

[144] Adam Zemla. LGA: a method for finding 3D similarities in protein structures. *Nucleic Acids Research*, 31(13):3370–3374, 07 2003. ISSN 0305-1048. doi: 10.1093/nar/gkg571. URL https://doi.org/10.1093/nar/gkg571.