Predicting and Reconstructing Proton NMR spectrum Exclusively with Structural Formula Using Bayesian Deep Learning Technique.

Hanwen Wang and Xinling Yu

CONTENTS

Hanwen Wang is a first year master in AMCS graduate group, University of Pennsylvania, PA 19104, USA mailto:wangh19@sas.upenn.edu

Xinling Yu is a first year master in AMCS graduate group, University of Pennsylvania, PA 19104, USA mailto:xlyu@sas.upenn.edu

***Abstract*— In this report we discuss about the use of machine learning models in the prediction of the proton NMR (Nuclear Magnetic Resonance) spectrum of hydrocarbon. With the combination of the Graphical Convolutional Neural Network and the Bayesian Neural Network, the model was able to predict chemical shift of certain atom basing on its local structure with an acceptable accuracy. And by using the Bayesian method in the neural network, the model was able to provide uncertainty via a production of probabilistic distribution to the result, and hence recognize the "noisiness" of the measurements due to possible factors, such as power stability, maintenance conditions, Helium quality, etc., that are inevitably unstable in the manual operation of the NMR instrument.**

I. INTRODUCTION

Nuclear Magnetic Resonance (NMR) is an information rich, non-destructive analytical technique that provides fine detailed information about the molecular structure, composition, and chemical dynamics. It is widely used in areas not limited to chemistry research, but also in related industries such as forensic sciences, food, health and even mining industries.

When the analyte (the substance pending analysis) is placed into the instrument, the strong constant magnetic field is then perturbed by a weak oscillating magnetic field[11], which will be responded by the resonance of the nuclei whose frequency characteristics match with that of the weak perturbation. Proton NMR, or more commonly referred as H^1 NMR, is a popular subfield of the NMR family with the above mechanism being applied to the hydrogen atoms exclusively. In this case, the characteristics of the proton is largely determined by the structure, and mostly the local structure around the target proton (which will be referred to later as "chemical environment"). Hence proton within different chemical environments can be identified with different resonance frequency (later referred as "chemical shift"), after an application of Fourier transformation. In practice, NMR instruments may have different specifications, which produces slightly differed result. For example, the company Bruker® provides NMRs whose operating frequency range from 300Hz to 800Hz. The unaffordable price of such instrument results in the extensive individual operating lifespan, and the actual implementation, such as the choice of solvent, operating pressure and temperature, or even different locations in geomagnetic, could also exacerbate the magnitude uncertainty in the data. Hence

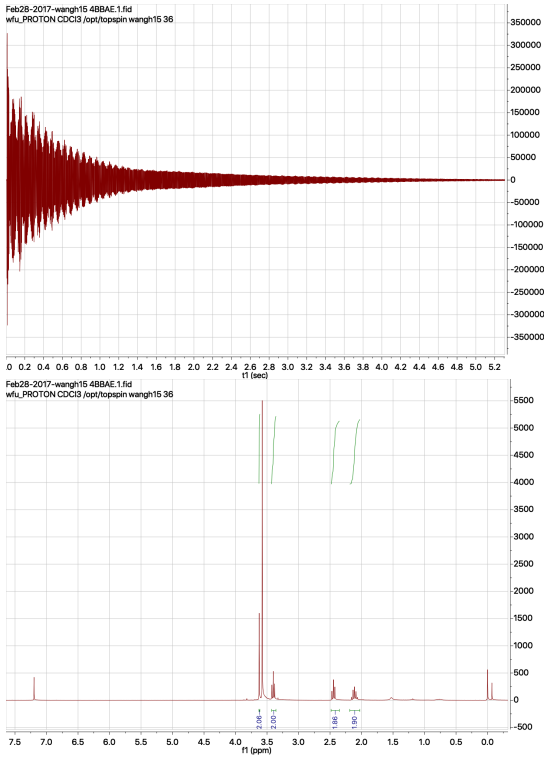


Fig. 1: (Top) Free Induction Decay (FID), (bottom) Fourier transformed spectrum. FID is the raw data collected with minimal processing, and the transformed spectrums are human interpretable. The height of the peak (intensity) is proportional to number of proton in the corresponding chemical environment.

although the frequencies are unique in theory for protons, it is likely that a number of conflicts exist. Such intrinsic discrepancy of data will require special handling shown later in this report to solve.

The proton NMR spectrum, especially those containing aromatic rings (such as benzene ring), could be difficult to interpret. The conventional prediction of chemical shift remains largely linear. A base chemical shift, usually determined by the elements, is corrected by addition or subtraction according to the presence of nearby functional groups, with a set of empirical rules. However, such rules fail when the complexity of the molecules, e.g. as in Figure 2 increases, and unsolvable conflicts may occur to crash the program or frustrate the researcher.

The peak assignment could be done using experimental methods such as 2-D NMR, e.g., COSY, HMBC, HSQC (see figure 3 to have an idea of how the 2-D spectrum). But again, extracting information from such increasingly complicated high dimensional graph could be as cumbersome.

Figure 4 shows a scenario that we wish to solve with the deep learning model. Like the one displayed in figure 2, conjugation system shown in 4 often brings ambiguities and conflicts in the application of the empirical rules mentioned

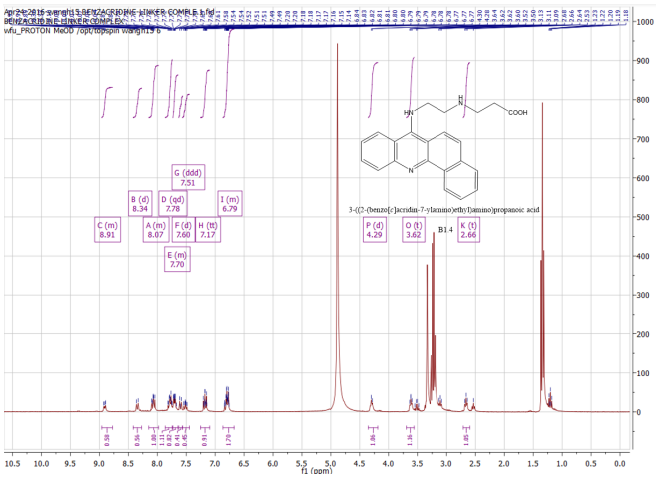


Fig. 2: The proton NMR sp..ectrum of an aromatic compound 3-((2-(benzo[c]acridin-7-ylamino)ethyl)amino)propanoic. The peaks in the left side region correspond to the proton on the aromatic rings. Differentiating those peaks from the visually similar chemical environments can be strenuous.

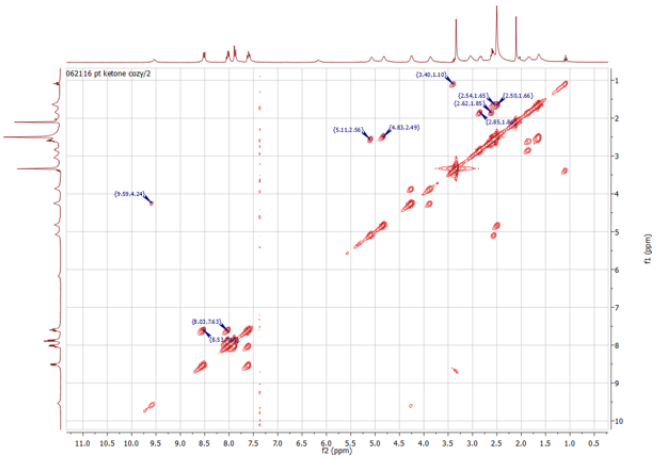


Fig. 3: 1H-1H COSY (COrrelated SpectroscopY). Nearby hydrogens (usually separated by less than 3 bonds, i.e., hydrogens attach on adjacent carbons) will show coupling effect, which results in peaks in the 2-D diagram. Although such high dimensional spectrum provides extra dimension of information, the interpretation could still be cumbersome.

above.

Fortunately, the rapidly developing machine learning methods gives us an opportunity to derive computer-interpretable rules from the beginning. Deep learning tools have been successfully applied to many domains, such as health-care, cheminformatics and computational mechanics. Most of the existing models [6] that predict the spectrum using deep learning concept rely heavily on the already obtained structure data from measurement such as X-ray diffraction, which leads to a contradiction that, the purity required for such prior characterizations has already exceeded the requirement for a reliable H^1 NMR spectrum. Hence these models are not applicable to the common laboratory practice where the

Fig. 4: Conjugating system. (Left) A linear conjugation system. (Right) a benzene ring, the most common circular conjugation system (also called aromatic system). Conjugation forms when double bonds and single bonds are placed intermittently. The Electron clouds overlap and delocalizes to form molecular orbitals. The conjugation offers unique features such as conductivity, florescence, polymerizability, and is hence widely used in a variety of industries. Most of the biomolecules, such as ribonucleic acid, deoxyribonucleic acid, and protein, also possesses functionalities that are closely related to conjugation system.

researchers only know the structural formula.

Other problem with the existing model is that the inputs of the models (such as SMILE[9] string) are rather nomenclatural than structural, which does not utilize all of the prior information on hand. They are also only point estimators that produce heavily degenerated spectrum[8], where the uncertainty in measurement is not accounted. These models may be very helpful when the structures of the large scale chemical are determined thoroughly and accurately, but chance that such situation occurs in daily organic synthesis is low.

Instead, our model aims at maximizing the information learned solely by structure formula without any prior characterization so that they can be used as a light weighted and cheap reference to individual researchers in small labs who are frustrated by the lack of expensive instruments. The accuracy may not be as high as the existing models that use actual structural data, but the Bayesian models[1] that are proposed to include will offer a mathematically grounded framework to reason about model uncertainty, so that the spectrum predicted will be more realistic and robust against subtle perturbations in practice. This project will also explore a new approach to digitize the structural related chemistry for other cheminformatical applications such as reaction design and product composition prediction.

II. Data Collection and Preprocessing

Given that the cheminformatics is still a field under developing, the data about the topological structural of the molecules is rare. And the speciality of NMR data acquisition also adds the difficulty of the collection of a training set. All of the data was cured manually from the AIST database[10]. The structural information is represented as bonding matrix. Given an indexing of the non H atoms, the entry $A_{i,j}$ of the bonding matrix A takes value from $0, 1, 2, 3, ...$, corresponding to none bond, single bond, double bond, triple bond, etc., if there exists other non carbon atoms with higher valence number, between atoms i, j.

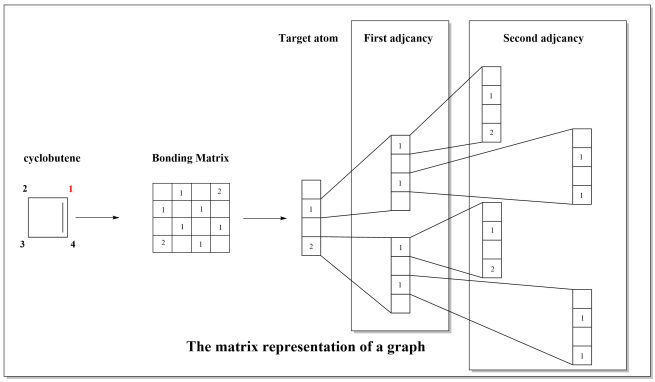


Fig. 5: A traverse on the backbone from atom 1 (red) represented in the bonding matrix.

III. Method

A. Graphical Convolutional Neural Network for Atomic Finger Print (GCNNAFP)

1) Assumptions: Before the introduction, a few assumptions have been made in advance to simplify the complicated reality that we are facing. These assumptions will also justify our choice of model architecture.

a) Assumption 1:: The chemical environment of the characterized H atom is solely based on the atom that it is bonded to. Hence we are not considering the enantiomers, geometric isomers, etc, although in proton NMR spectroscopy these isomers only differ slightly. We willB discard the H atom and only study the backbone system.

b) Assumption 2:: The influence of remote atoms in the molecule exceeding certain convolutional depth is negligible. Hence we ruled out the rare cases in which complex long range couplings, such as "quenching effect" of heavy metal atoms, are involved.

c) Assumption 2:: All the peaks are singlelet, i.e., every chemical environment corresponds to a single scaler of chemical shift. This assumption is a compromise to the lack of data on the coupling effect that produces peak splitting.

d) Assumption 3:: The information only flows through bonds. This is a crucial assumption that allows us to only use bonding matrix as the input data, providing that the actual measured structural data is missing.

e) Assumption 4:: The height of the peak (intensity) is strictly proportional to the number of the proton on the non H atom whose chemical environment is predicted, so that normalization is not needed.

2) Introduction: Inspired by [5], we used a GCNN (Graphical Convolutional Neural Network) to characterize the chemical environment of a given atom. GCNN shares a few similarities with the Convolutional Neural Network (CNN). They both utilize on 1) local connectivity, 2) parameter sharing, and most importantly, 3) invariance to certain transformations such as shifting and flipping, to

recognize patterns and features[7].
However, among all the applications, CNN is most popularly operated on "regular" data[7], such as grids (images) and sequences (text), which is not applicable in out situation since an atom belong to certain molecule could have different connectivity, such as numbers, classes, or even the directions of the bonds that it receives or sends. Hence, GCNN, as a generalization of the CNN, becomes a solution to our scenario. GCNN has the advantage of being able to operate on nodes and edges whose connectivities is not uniform throughout the whole graph, and its ability to remain invariant against permutation of indexing of the nodes and edges, when the graph is represented from visually to mathematically. Facing the latter case, the conventional solution is to apply full a permutation to each data points, which could result in a size factorial growth in size and hence training time and storage space.

The ultimate purpose is to use bond-specific functions to learn actual structural features such as sp-hybridizations, relative bond lengths and bond angles. These are [the most] determining factors of the local chemical environment.

3) Mathematical formulation: For a length N chain of non H atoms, let $v_0, v_i^{(j)} \in \mathbb{R}^{N_v}, i = 1, ..., N, j = 1, ..., M$ be the atomic feature vectors of atoms where v_0 is that of the target atom, and $v_i^{(j)}$ is the i-th closest non H atom to the target atom in j-th chain. $v_0 = v_i^{(j)}$ in this case since we only consider hydrocarbons. Let $w_0, w_i^{(j)} \in \mathbb{R}^{N_e}$ be the chemical environment vectors defined similarly. $w_N^{(j)} = \mathbf{0}$ as the end of the chain. $b_i^{(j)} \in \mathbb{R}^3$ is the bond between i and $i-1$ atoms of j-th chain. In this case $b_i^{(j)}$ takes value from $1, 2, 3$ for single, double and triple bond respectively. Two reduce the number of trained parameters, neural network functions $f_k : \mathbb{R}^{N_v+N_e+3} \mapsto \mathbb{R}^{N_e}$ are designed to predict how information flows through single, double and triple bonds, respectively. A convolution of depth N on an atom A is formulated as

$$F_N(A) = \sum_{j=1}^{M} \bigcirc_{i=1}^{N} \left(f_{b_i^{(j)}}(w_i^{(j)}), v_i^{(j)} \right), \tag{1}$$

where

$$\bigcirc_{i=1}^{N} \left(f_{b_i^{(j)}}(w_i^{(j)}), v_i^{(j)} \right) = f_{b_1^{(j)}} \left(\bigcirc_{i=2}^{N} \left(f_{b_i^{(j)}}(w_i^{(j)}), v_1^{(j)} \right) \right) = \cdots \tag{2}$$

the function composition, is defined recursively. Note that

$$w_K^{(j)} = f_{b_{K+1}^{(j)}} \left(\bigcirc_{i=K+1}^{N} \left(f_{b_i^{(j)}}(w_i^{(j)}), v_{K+1}^{(j)} \right) \right), \tag{3}$$

is a intermediate step of information flow from the end of $j-$th chain to $K-$th closest atom on $j-$th chain. Such functional composition is often referred to as message passing or neighborhood. It transfers, processes, and nonlinearly sums the information from the far end. Depending on the scenes, simple Recursive Neural Network (RNN) or even the slightly complicated version, Long Short Term Memory networks (LSTM) can be used as aggregators[7] as well. In our scenario, since the graph is often small (with only less then 20 nodes), we just use simple neural network to define our aggregator function f_ks.

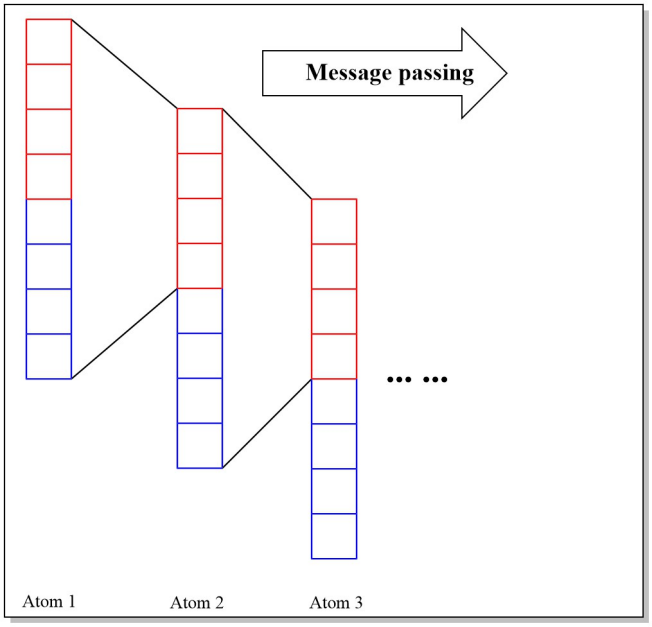


Fig. 6: Neighborhood aggregation scheme. Red rectangles are vectors containing information flowed to the intermediate atoms, and blue rectangles are intrinsic atomic information.

The parameters in f_ks are updated with a unsupervised training fashion. Although properly initialized neural networks with randomized weights have already proven[5] to have the ability to separate different chemical environments, we trained the GCNNAFP with negative mean square error as loss function to induce separation of different chemical environments in the latent space. Separation of the training of GNNAFP and the later neural network also prevents the optimization from entering the "flat region" of the loss function and can hence promotes better convergence and accuracy.

a) Implementation: Given the dataset $\boldsymbol{A}$ consists of N bonding matrices whose size does not necessary agree, the forward pass with depth 3 convolution to some individual bonding matrix A is implemented as Algorithm 1.

B. Bayesian Neural Network (BNN)

1) Introduction: Deep neural networks are connectionist systems that learn to perform tasks by learning on examples without having prior knowledge about the tasks. Despite Neural Networks architectures achieving state-of-the-art results in almost all classification and regression tasks, neural networks still make over-confident decisions. A measure of uncertainty in the prediction is missing from the current Neural Networks architectures. Very careful training, weight control measures like regularization of weights and similar techniques are needed to make the models susceptible to over-fitting issues. Research over the past few years has made

initialEnv = zeros(1,environmentFeature)
envVector = zeros([numOfAtom,BondFeatureDim,3])
for $n \in$ *enumerate(columns of)*A **do**
  for $i \in \{1, ..., numOfAtom\}/\{n, A_{n,i} = 0\}$ **do**
    firstFeature = $f_{A_{i,n}}$(cat(initialEnv,atomFeature$_{i,n}$))
    envVector[n,:,1] = envVector[n,:,0] + firstFeature
    for $j \in \{1, ..., numOfAtom\}/\{n, i, A_{j,i} = 0\}$ **do**
      firstFeature = $f_{A_{i,j}}$(cat(initialEnv,atomFeature$_{i,j}$))
      secondFeature = $f_{A_{i,n}}$(cat(firstFeature,atomFeature$_{i,n}$))
      envVector[n,:,2] = envVector[n,:,0] + secondFeature
      for $k \in \{1, ..., numOfAtom\}/\{n, k, 1, A_{k,j} = 0\}$ **do**
        firstFeature = $f_{A_{j,k}}$(cat(initialEnv,atomFeature$_{j,k}$))
        secondFeature = $f_{A_{i,j}}$(cat(firstFeature,atomFeature$_{i,j}$))
        thirdFeature = $f_{A_{i,n}}$(cat(second,atomFeature$_{i,n}$))
        envVector[n,:,3] = envVector[n,:,0] + thirdFeature
      end
    end
  end
end
return flatten(Vc, retain_dim = 1)

Algorithm 1: A forward pass of GCNNAFP that returns a list of atomic finger print. Four nested for loops search for paths up to depth 3 with non-repeating vertices.

progress towards efficient methods of obtaining uncertainty from deep models, here we will explore the basics of Bayesian deep learning and use the Bayes by Backprop algorithm [4] to recover the uncertainty of prediction of the chemical shift.

2) Mathematical formulation: In the probabilistic view of a neural network $P(\mathbf{y}|\mathbf{x}, \mathbf{w})$: given an input $\mathbf{x} \in \mathbb{R}^d$, for the fixed weights $\mathbf{w}$ (or the distribution of the weights) learnt from the training data, the neural network gives a probability to each possible output.
Given the training data $\mathcal{D} = (\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$, using bayes rule, we can represent the weights $\mathbf{w}$ of a neural network as:

$$P(\mathbf{w}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{P(\mathcal{D})} = \frac{P(\mathcal{D}|\mathbf{w})P(\mathbf{w})}{\int P(\mathcal{D}|\mathbf{w})P(\mathbf{w})\mathrm{d}\mathbf{w}}, \quad (4)$$

here $P(\mathbf{w}|\mathcal{D})$ is the posterior probability of $\mathbf{w}$, which is the probability distribution over the weights, given the data set $\mathcal{D}$. $P(\mathcal{D}|\mathbf{w})$ represents the likelihood of $\mathbf{w}$, which is a probability distribution over the data, given a fixed setting of weights.$P(\mathbf{w})$ is the prior probability of the weights. $P(\mathcal{D})$ is the marginal likelihood of the data.
In the frequentist perspective, we usually train neural networks by updating weights using gradient descent seeks to find the weights which best explain the data. This can be viewed as learning the weights which maximize the likelihood $P(\mathcal{D}|\mathbf{w})$ by Maximum Likelihood Estimation (MLE):

$$\begin{aligned} \mathbf{w}^{\mathrm{MLE}} &= \arg\max_{\mathbf{w}} P(\mathcal{D}|\mathbf{w}) \\ &= \arg\max_{\mathbf{w}} \prod_i P\left(y_i|x_i, \mathbf{w}\right), \end{aligned} \quad (5)$$

here during the computation of the likelihood, the weights of our model are fixed, but the data is viewed as a random variable.
In the bayesian perspective, instead fix the weights, we fix the data and view weights as a random variable, then we can train neural networks by maximizing the posterior probability $P(\mathbf{w}|\mathcal{D})$ via maximumizing a posterior (MAP) learning:

$$\begin{aligned} \mathbf{w}^{\mathrm{MAP}} &= \arg\max_{\mathbf{w}} P(\mathbf{w}|\mathcal{D}) \\ &= \arg\max_{\mathbf{w}} P(\mathcal{D}|\mathbf{w})P(\mathbf{w}) \\ &= \arg\max_{\mathbf{w}} \mathrm{mog}\, P(\mathcal{D}|\mathbf{w}) + \log P(\mathbf{w}), \end{aligned} \quad (6)$$

the bayesian inference of neural networks is the process of computing he entire posterior distribution $P(\mathbf{w}|\mathcal{D})$. This distribution answers predictive queries about unknown data by taking expectations: the predictive distribution of an unknown label $\hat{\mathbf{y}}$ of an input test data $\hat{\mathbf{x}}$ is given by:

$$\begin{aligned} P(\hat{\mathbf{y}}|\hat{\mathbf{x}}) &= \mathbb{E}_{P(\mathbf{w}|\mathcal{D})}[P(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \mathbf{w})] \\ &= \int P(\hat{\mathbf{y}}|\hat{\mathbf{x}}, \mathbf{w})P(\mathbf{w}|\mathcal{D})\mathrm{d}\mathbf{w}, \end{aligned} \quad (7)$$

each possible configuration of the weights, weighted according to the posterior distribution, makes a prediction about the unknown label given the test data item $\hat{\mathbf{x}}$.

a) Variational Inference: However, while $P(\mathbf{w})$ is the prior we can choose and $P(\mathcal{D}|\mathbf{w})$ is the likelihood we can directly compute from the data, the computation of $P(\mathcal{D})$ is problematic, so we need to use approximations of $P(\mathbf{w}|\mathcal{D})$ in order to make Bayesian predictions. Variational inference constructs a new distribution $q(\mathbf{w}|\theta)$ over the weights $\mathbf{w}$ and parameterized by θ, and uses the distribution to approximate $P(\mathbf{w}|\mathcal{D})$ (finds the θ) by minimizing the Kullback-Leibler (KL) divergence with the true Bayesian posterior $P(\mathbf{w}|\mathcal{D})$:

$$\begin{aligned} \theta^* &= \arg\min_{\theta} \mathrm{KL}[q(\mathbf{w}|\theta)\|P(\mathbf{w}|\mathcal{D})] \\ &= \arg\min_{\theta} \int q(\mathbf{w}|\theta) \log \frac{q(\mathbf{w}|\theta)}{P(\mathbf{w})P(\mathcal{D}|\mathbf{w})}\mathrm{d}\mathbf{w} \\ &= \arg\min_{\theta} \mathrm{KL}[q(\mathbf{w}|\theta)\|P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})], \end{aligned} \quad (8)$$

we denote the cost function, which is known as the variational free energy [3] as:

$$\mathcal{F}(\mathcal{D},\theta) = \mathrm{KL}[q(\mathbf{w}|\theta)\|P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})], \tag{9}$$

the first term of the cost function intends to make the variational posterior that is close to the prior we choose, the second term of the cost function intends to explain the complexity of the data well.
Actually, $-\mathcal{F}(\mathcal{D},\theta)$ is also known as expected lower bound (ELBO) [2], thus when we minimize our cost function, we are also maximizing the ELBO.

b) Backpropagation: Computing the expectation of the likelihood over the variational posterior directly is very difficult, so again we use approximate the cost function as:

$$\mathcal{F}(\mathcal{D},\theta) \approx \sum_{i=1}^{n} \log q\left(\mathbf{w}^{(i)}|\theta\right) - \log P\left(\mathbf{w}^{(i)}\right) - \log P\left(\mathcal{D}|\mathbf{w}^{(i)}\right), \tag{10}$$

where $\mathbf{w}^{(i)}$ denotes the i th Monte Carlo sample drawn from the variational posterior $q\left(\mathbf{w}^{(i)}|\theta\right)$. When using automatic differentiation as provided by frameworks such as PyTorch, we only need to worry about implementing this sampling, and setting up the cost function as above, and can leverage our usual backpropagation methods to train a model.

3) Implementation:

a) Problem Setup: Given the training data $\mathcal{D} = (\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$, where $\mathbf{x_i}$ is the vectorized representation of chemical environment, and $\mathbf{y_i}$ is the measured chemical shift (horizontal position of the peak) and the prior of the weights $\mathbf{w}$ of the neural network (we used Gaussian prior).

b) Goal: Given a new input $\hat{\mathbf{x}}$, the goal is to find the predictive distribution of the chemical shift $\hat{\mathbf{x}}$ via:

$$\begin{aligned} P(\hat{\mathbf{y}}|\hat{\mathbf{x}}) &= \mathbb{E}_{P(\mathbf{w}|\mathcal{D})}[P(\hat{\mathbf{y}}|\hat{\mathbf{x}},\mathbf{w})] \\ &= \int P(\hat{\mathbf{y}}|\hat{\mathbf{x}},\mathbf{w})P(\mathbf{w}|\mathcal{D})\mathrm{d}\mathbf{w}, \end{aligned} \tag{11}$$

here the posterior distribution $P(\mathbf{w}|\mathcal{D})$ is approximated by variational posterior distribution $q(\mathbf{w}|\theta^*)$, and θ^* is given by minimizing the approximating negative ELBO cost function[2]:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} \log q\left(\mathbf{w}^{(i)}|\theta\right) - \log P\left(\mathbf{w}^{(i)}\right) - \log P\left(\mathcal{D}|\mathbf{w}^{(i)}\right) \tag{12}$$

here $\mathbf{w}^{(i)}$ denotes the i th Monte Carlo sample drawn from the variational posterior $q\left(\mathbf{w}^{(i)}|\theta\right)$.

The process is described as Algorithm 2 below:

Figure 7 shows how the model combines GCNNAFP and BNN.

Input: Training set $\mathcal{D} = (\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$, vectorized representation of the chemical environment $\hat{\mathbf{x}}$, iterations, learning rate = α, number of samples = N;
Initialize the posterior weights: The initial variational posterior parameters are $\theta = (\mu, \rho)$, where μ and ρ are sampled from uniform distributions and the posterior weights $\mathbf{w} \sim \mathcal{N}\left(\mathbf{w}|\mu, \rho^2\right)$;
while *i < iterations* **do**
  Sample a parameter free noise $\epsilon \sim \mathcal{N}(0, I)$;
  Let $\mathbf{w} = \mu + \log(1 + \exp(\rho)) \circ \epsilon$;
  Let $\theta = (\mu, \rho)$;
  `/* KL-loss */`
  Let $f(\mathbf{w}, \theta) = \log q(\mathbf{w}|\theta) - \log P(\mathbf{w})P(\mathcal{D}|\mathbf{w})$;
  Calculate the gradient with respect to the mean $\Delta_\mu = \frac{\partial f(\mathbf{w},\theta)}{\partial \mathbf{w}} + \frac{\partial f(\mathbf{w},\theta)}{\partial \mu}$;
  Calculate the gradient with respect to the standard deviation parameter$\Delta_\rho = \frac{\partial f(\mathbf{w},\theta)}{\partial \mathbf{w}} \frac{\epsilon}{1+\exp(-\rho)} + \frac{\partial f(\mathbf{w},\theta)}{\partial \rho}$;
  Update the variational parameters:

$$\mu \leftarrow \mu - \alpha\Delta_\mu$$
$$\rho \leftarrow \rho - \alpha\Delta_\rho$$

end
Output 1: $\theta^* = (\mu^*, \rho^*)$;
while *i < M* **do**
  Sample $\boldsymbol{w}$ from $\mathcal{N}\left(\mu^*, (\rho^*)^2\right)$; Calculate $\hat{\mathbf{y}}_j$ via $\hat{\mathbf{x}}$ and the neural network parameterized by $\mathbf{w}$
end
`/* Predictive distribution */`
$\hat{\mathbf{y}} = (\hat{\mathbf{y}}_j)_{j=1}^M$
Output 2: predictive distribution = gaussiankde($\hat{\mathbf{y}}$)

Algorithm 2: The training of a Bayesian Neural Network. The model outputs an estimated distribution instead of a value.

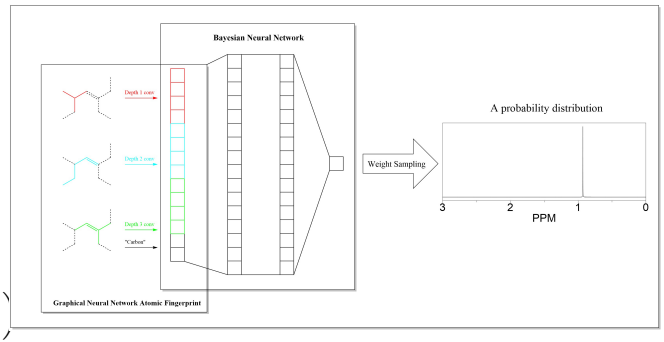


Fig. 7: Schematic workflow. The vertices of the structural formula are non H atoms, and the edges are the bonds. The dashed lines show edges are not covered in each convolution.

IV. Result and Discussion

Table I shows the choices of hyper parameters. The convolution depth 4 is the minimal depth that can detect

the existence of a nearby benzene ring, which often causes confusion when a human interprets the spectrum. The choice of other parameters follows standard neural network practice.

Variables	Detail	Parameter
depth	Convolution depth	4
bond_feature	chemical environment feature vector	8
atom_feature	atom feature vector	8
f_1, f_2, f_3	Aggregator	$(8+8)\times 16\times 16\times 8, \tanh$
BNN	Bayesian Neural Network	$(4\cdot 8+8)\times 128\times 1$
(μ, ρ^*)	Normal prior	$(0, 0.1)$

TABLE I: Choice of hyper parameters. Due to the complexity of the parameter updating in the Bayesian Neural Network and the limited computational resource, we have to compromise the accuracy of the model to improve performance.

A. GCNNAFP

Figure 8 shows the effect of separating the trainings of the GCNNAFP and subsequent fully connected layers. We can see that the separation brought improvement in convergence rate while the accuracy remains almost unchanged.

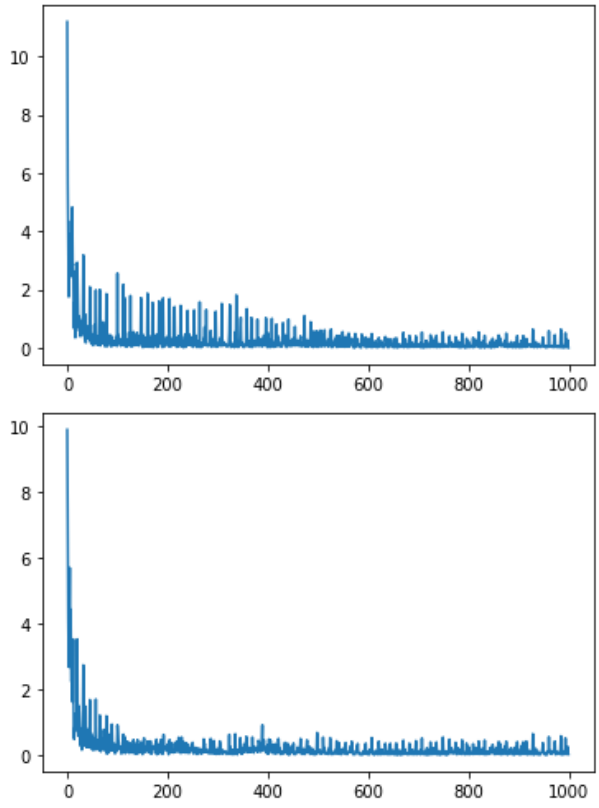


Fig. 8: (Top) simultaneous update of GCNNAFP and subsequent fully connected layers. (Bottom) separated trainings of GCNNAFP and subsequent fully connected layers. In this case the fully connected layers include two hidden layers with 128 neurons each. The two models return similar mean square error ~ 0.22 on the test set.

B. BNN

Although the BNN is proposed in figure 7 as taking both the chemical environment and atomic feature as variable, in practice we found that since the atoms are exclusively carbon, the defacto "collapsed" dimension induces a ill posed condition for numerical computation. Hence we removed the atomic feature to ensure that the program runs with full functionality. Figure 9 shows the evolution of the distribution over training epochs. Theoretically, the distribution should converge almost surely to the true posterior distribution. However, we notice that the mode "sweeps" through the true value, while the variance continues to decrease. This implies that the number of training epochs trades mode\mean accuracy with the variance.

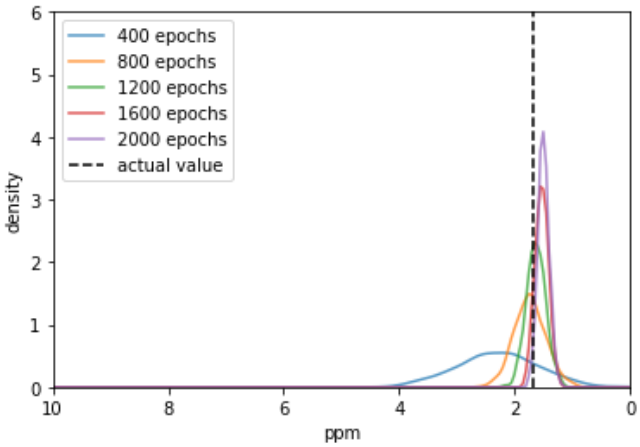


Fig. 9: The update of distribution of one prediction over training epochs. Although the probability density function continues to grow around its posterior mean, signs of almost sure convergence does show up.

The overall model provides an accurate prediction on most compounds. After being trained for 4000 epochs, the predicted chemical shift on the test set has an accuracy of mean squared error ~ 0.17 on individual atoms. However, when the results from one compound are combined together, compounds having a common feature of exhibiting a tree structure where cycle is not formed usually displays a better visual accuracy. Figure 10 shows a successfully prediction on cis-1,2,4-hexatriene. The model was able to learn the effect of the conjugate system brought by the coupling of several double bonds. The model behaves not as well as above on the prediction of compounds that have complicated topological structure, e.g., shown as in figure 11. The prediction suffers from a lack of training data on large aromatic compound. Note in real case the conjugation system of benzene ring, if correctly predicted, should only has two peaks in the aromatic region (in the left of the spectrum), since all the bonds are in an identical state between singe and double bond. Due to the limit of training data the model was not able to learn that rule yet. The individual peak nevertheless does show a correct order of separation, and is hence still valuable when used as a reference. Basing on the successful extraction of the conjugation feature in figure 10, we believe that after provided with sufficient samples, we should be able

to considerably improved the model.

Figure 12 shows the influence of the choice of the prior. Shaper prior distribution may result in an overall better behavior, while a flatter prior may have better performance of its mean, if chosen as a deterministic estimator. An extreme case is figure 12d where the prior dominates.

Figure 13,14 show that the error accumulates near chemical shift of 7.0 ppm and 1 ppm, where are the regions for protons on aromatic system and alkyne end, respectively. Both of the two classes of compound are absent in the collected dataset, leading possibly towards greater error than the others.

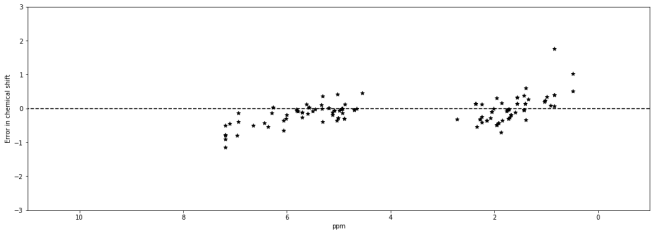


Fig. 13: Distribution of errors.

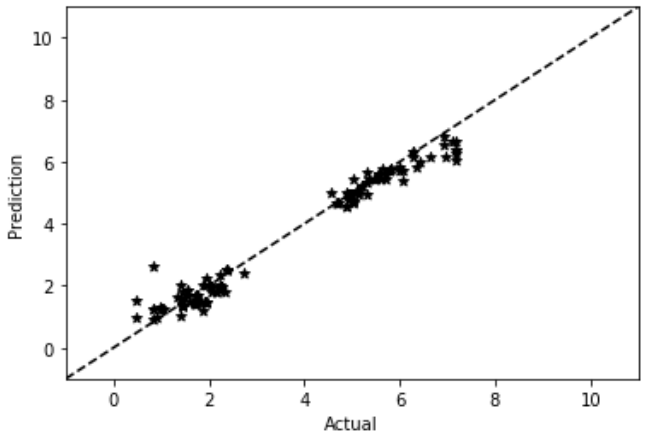


Fig. 14: Prediction versus actual data.

V. Summary

The model successfully extracts local structural feature of the molecules and was able to make satisfiable prediction with quantified uncertainty. Even in the case where the reconstructed spectrum does not resemble the original spectrum, the correct order of the peak still offers valuable information for peak assignment. Further higher dimensional characterization can be tuned to focus on certain region of the spectrum with high uncertainty presence to save time and fund cost.

VI. Future improvement

In this case the compounds are exclusively hydrocarbon that only has hydrogens and carbons in their composition. Due to the difficulty of manual data curation, the size of the data is also limited. In the future, we hope to use spyder in Python to automatically collects compounds with a variety of structural feature such as coordinate bonding, conjugated aromatic rings, and heavy metals. The model can also be improved by using more efficient aggregators and training goal in the GCNNAFP part to reduce the uncertainty induced during the encoder training. A more carefully chosen prior in BNN part will also help to solve the trade off between point-estimator accuracy and variance of the estimator.

Basing on the prediction result, a decision theoretic criterion can be made to determine the estimated ordering of the peaks, which could be used as a reference to peak assignments after the acquisition of real spectrum.

VII. Acknowledgment

The essential technological part of this project is supported [partially] by the Spring 2020 Technology Grant distributed by the Graduate Center at Penn.

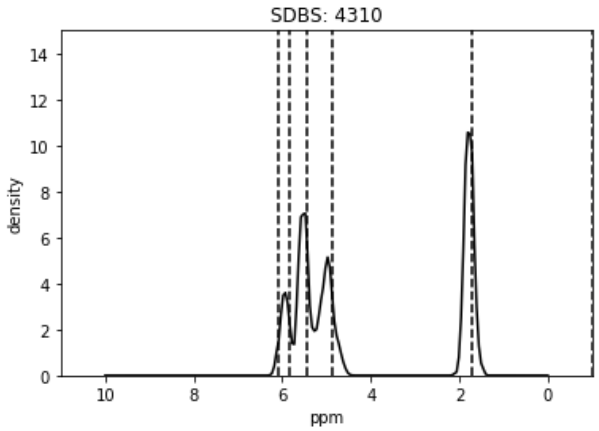
We would also like to show our gratitude to professor Dr. Perdikaris and teaching assistances Yibo Yang and Taruna Kar, for their continued support through such unprecedented COVID-19 pandemic.

National Institute of Advanced Industrial Science and Technology (AIST) for provides the free access to their Spectral Database for Organic Compounds (SDBS)[10], from which all our data are collected.

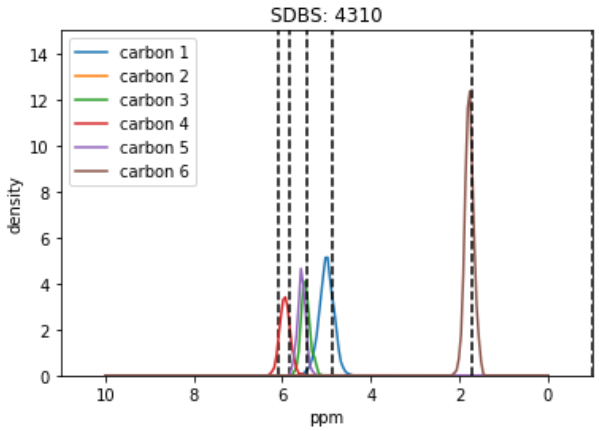
VIII. Supplemental material

All the files, including scripts, raw data, atomic fingerprint data, model state dictionary has been uploaded to Github repository https://github.com/xlyu0127/ENM-531-Project

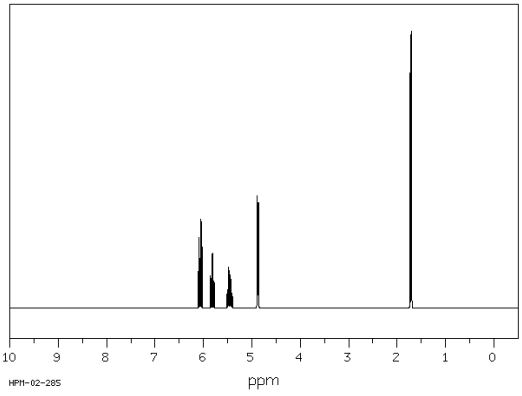
`traindata`, `testdata` are the raw data whose first column is the SDBS number of the compound, second column is the bond matrix, third column is the chemical shift (labeled to -1 if no hydrogen is attached). `r4testcode`,`r4testY`, `r4traincode`, `r4trainY` are test and training data of BNN with the -1 entries being removed. `MGNNEMVreverse.py` and `bnn2.py` are script files containing class and function definitions for GCNNAFP and BNN. `encode_result.py` generates atomic fingerprint from raw data to atomic finger print, after the state dictionary `r4e` is loaded for GCNNAFP. Commented blocks in `full4r` generate most of the plots.



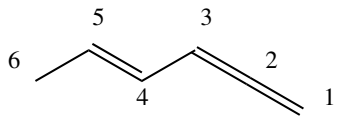
(a) Recovered spectrum.



(b) Recovered spectrum of individual peak.

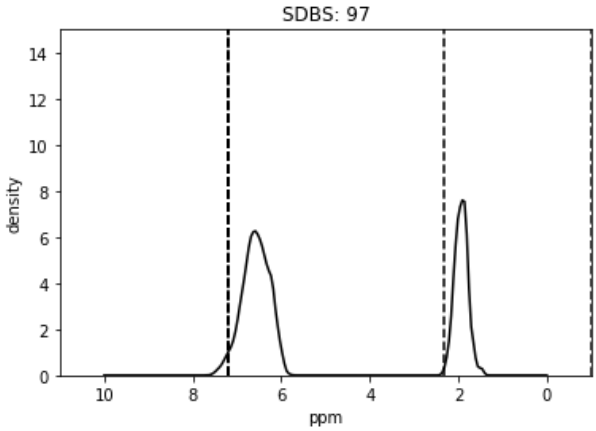


(c) Real spectrum.[10]

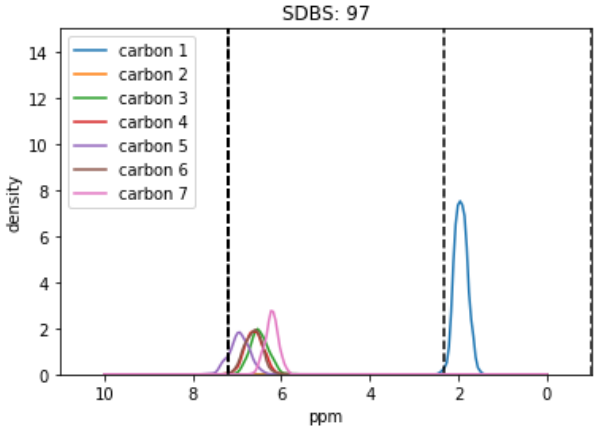


(d) Structure of the analyte cis-1,2,4-hexatriene.

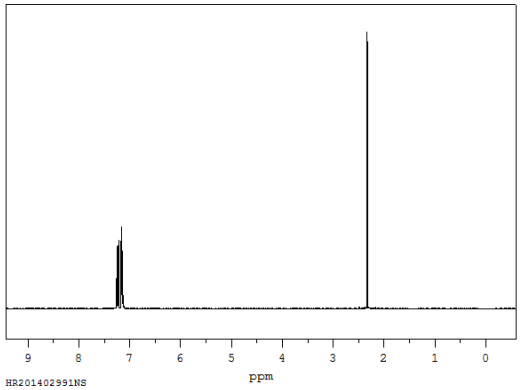
Fig. 10: The predicted spectrum of compound cis-1,2,4-hexatriene. The dashed lines are the degenerated frequencies used as data input.



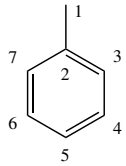
(a) Recovered spectrum.



(b) Recovered spectrum of individual peak.

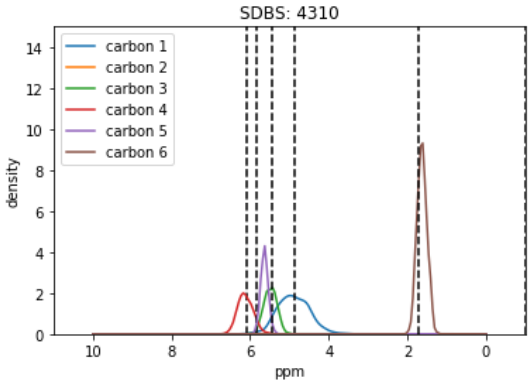


(c) Real spectrum[10]

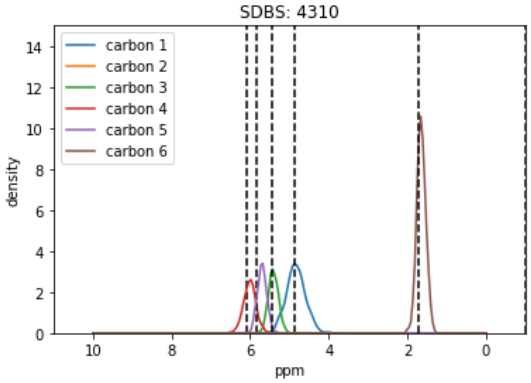


(d) Structure of the analyte methylbenzene.

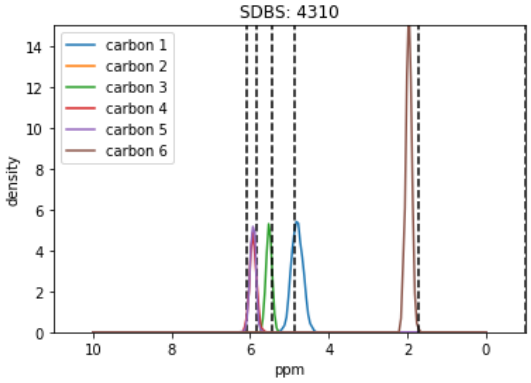
Fig. 11: The predicted spectrum of methylbenzene. The dashed lines are the degenerated frequencies used as data input.



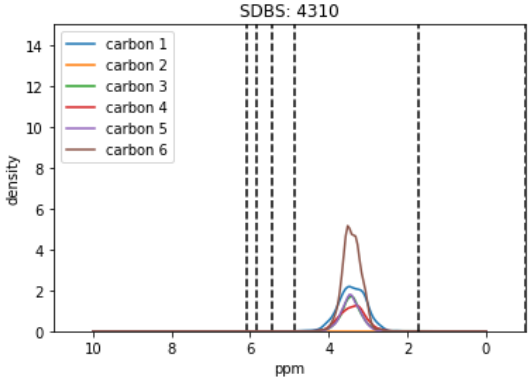
(a) $\sigma^2 = 1$



(b) $\sigma^2 = 0.01$



(c) $\sigma^2 = 0.001$



(d) $\sigma^2 = 0.0001$

Fig. 12: The effect of prior parameters. The variances of prior normal distribution is tuned to observe difference.

References

[1] Radford M. Neal. *Bayesian Learning for Neural Networks*. en. Ed. by P. Bickel et al. Vol. 118. Lecture Notes in Statistics. New York, NY: Springer New York, 1996. ISBN: 978-0-387-94724-2 978-1-4612-0745-0. DOI: 10.1007/978-1-4612-0745-0. URL: http://link.springer.com/10.1007/978-1-4612-0745-0 (visited on 04/10/2020).

[2] Lawrence K Saul, Tommi Jaakkola, and Michael I Jordan. "Mean field theory for sigmoid belief networks". In: *Journal of artificial intelligence research* 4 (1996), pp. 61–76.

[3] Radford M Neal and Geoffrey E Hinton. "A view of the EM algorithm that justifies incremental, sparse, and other variants". In: *Learning in graphical models*. Springer, 1998, pp. 355–368.

[4] Charles Blundell et al. "Weight uncertainty in neural networks". In: *arXiv preprint arXiv:1505.05424* (2015).

[5] David K Duvenaud et al. "Convolutional Networks on Graphs for Learning Molecular Fingerprints". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes et al. Curran Associates, Inc., 2015, pp. 2224–2232. URL: http://papers.nips.cc/paper/5954-convolutional-networks-on-graphs-for-learning-molecular-fingerprints.pdf.

[6] Federico M. Paruzzo et al. "Chemical shifts in molecular solids by machine learning". en. In: *Nature Communications* 9.1 (Dec. 2018), p. 4501. ISSN: 2041-1723. DOI: 10.1038/s41467-018-06972-x. URL: http://www.nature.com/articles/s41467-018-06972-x (visited on 04/10/2020).

[7] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *arXiv preprint arXiv:1812.08434* (2018).

[8] Eric Jonas and Stefan Kuhn. "Rapid prediction of NMR spectral properties with quantified uncertainty". en. In: *Journal of Cheminformatics* 11.1 (Dec. 2019), p. 50. ISSN: 1758-2946. DOI: 10.1186/s13321-019-0374-3. URL: https://jcheminf.biomedcentral.com/articles/10.1186/s13321-019-0374-3 (visited on 04/10/2020).

[9] *Simplified molecular-input line-entry system*. en. Page Version ID: 944722910. Mar. 2020. URL: https://en.wikipedia.org/w/index.php?title=Simplified_molecular-input_line-entry_system&oldid=944722910 (visited on 04/10/2020).

[10] *Spectral Database for Organic Compounds, AIST*. https://sdbs.db.aist.go.jp/. [Online; accessed 12-May-2020]. 2020.

[11] Wikipedia contributors. *Nuclear magnetic resonance — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Nuclear_magnetic_resonance&oldid=952360469. [Online; accessed 12-May-2020]. 2020.