1. INTRODUCTION

Vibrational spectroscopy is a powerful tool for the elucidation of molecular structures. It allows researchers to determine the molecular geometry by analyzing the vibrational modes of a molecule, which are sensitive to the positions of atoms relative to one another. Vibrational spectroscopy encompasses a range of techniques used to measure the vibrational energy levels of molecules, with the most common forms being infrared (IR) spectroscopy and Raman spectroscopy. In IR spectroscopy, the molecule absorbs infrared light, causing transitions between vibrational energy levels. Raman spectroscopy involves inelastic scattering of light, where incident photons are scattered at different energies due to vibrational transitions within the molecule. These spectroscopic techniques provide detailed information about the molecular structure, as the vibrational frequencies depend on the masses of the atoms and the strength of the chemical bonds. However, there are also finer details in the vibrational spectra due to differences in the environment of the functional groups. To analyze these subtle effects, quantum chemical calculations are required. By combining experimental spectroscopic data with computational methods, researchers can infer bond lengths, bond angles, and other structural parameters, leading to a comprehensive understanding of the molecular geometry and dynamics. Determining a reliable molecular structure typically involves geometry optimization, which accounts for a significant portion of computational resources—estimated to be between 60% and 85% of total CPU cycles in computational chemistry. Once a molecular geometry is optimized, it must be characterized to ensure it represents a true minimum energy state. This is achieved through vibrational analysis, where vibrational frequencies are computed and verified to be real values, confirming the stability of the structure. The traditional approach to correlating experimental spectra with molecular structures involves running numerous frequency calculations and manually checking which structure best matches the experimental data. This process is time-consuming and relies heavily on human interpretation, which can be influenced by subjective biases. Additionally, the information from spectra that do not match is often discarded, representing a significant loss of potentially valuable data.

To address these issues, we propose the application of machine learning (ML) to find a correlation between the differences in vibrational spectra and the differences in torsional angles of molecular structures. By leveraging ML, we aim to develop a model that can efficiently learn features from vibrational spectra and predict the corresponding molecular structures, thereby reducing computational time and improving the accuracy of structural determinations. In summary, while vibrational spectroscopy is a method for determining molecular structures, the current practice of using random structures and manual checks is inefficient. Our goal is to utilize ML to enhance the process by learning from the spectra, ultimately providing a more systematic and accurate approach to structure elucidation.

Codes : **https://github.com/anshug123/mitacs.git**

2. DATASETS AND PRE-PROCESSING

2.1 CREST Calculations

CREST (Conformer–Rotamer Ensemble Sampling Tool) was employed to explore the conformational space of the molecule and generate a comprehensive dataset of molecular conformers. CREST is a powerful tool designed for molecular structure exploration, incorporating various advanced algorithms and methodologies to sample conformers, optimize geometries, and analyze molecular structures.

CREST generated output files containing detailed information about each conformer, including:

- **Geometry Information**: The positions of atoms and bond lengths.
- **Torsional Angles**: Measurements of torsional angles for various atom groups, essential for understanding the conformational changes.

CREST was used to explore the conformational space of delta THC, generating 1326 conformers to generate torsion data.

2.2 Torsion Data

In the provided dataset, there are 19 specific torsion angles (t1 to t19) measured for each of the 1326 conformers of the molecule. Torsion angles measure the rotation around a bond axis between successive atoms, indicating how one part of the molecule twists relative to another. For example, a torsion angle specified as "TORSION ATOMS=14,2,6,7" quantifies the angular relationship among atoms 14, 2, 6, and 7. Each torsion angle ranges from -π to π. We made a comprehensive list of all possible torsions, removed those that are similar due to symmetry, and excluded those that did not make chemical sense (i.e., torsions that would remain constant regardless of the molecular conformation).

t1: TORSION ATOMS=14,2,6,7

t2: TORSION ATOMS=2,6,7,10

t3: TORSION ATOMS=6,7,10,13

t4: TORSION ATOMS=7,10,13,14

t5: TORSION ATOMS=10,13,14,2

t6: TORSION ATOMS=13,10,16,25

t7: TORSION ATOMS=10,16,25,20

t8: TORSION ATOMS=16,25,20,18

t9: TORSION ATOMS=25,20,18,13

t10: TORSION ATOMS=20,18,13,10

t11: TORSION ATOMS=18,30,52,53

t12: TORSION ATOMS=31,32,36,37

t13: TORSION ATOMS=32,36,37,40

t14: TORSION ATOMS=36,37,40,43

t15: TORSION ATOMS=37,40,43,46

t16: TORSION ATOMS=40,43,46,50

t17: TORSION ATOMS=14,2,1,3

t18: TORSION ATOMS=10,16,21,22

t19: TORSION ATOMS=10,16,26,27

	Torsion 1	**Torsion 2**	**Torsion 3**	**…**	**Torsion 19**
Conformer 1					
Conformer 2					
Conformer 3					
….					
Conformer 1326					

2.3 Preprocessing of Torsion Data

The torsion data in the dataset exhibits several notable characteristics. Firstly, it shows numerous duplicates where torsion angle values are closely clustered together, suggesting repetitive or similar molecular conformations. Moreover, many conformers demonstrate three-fold symmetry, which further contributes to the presence of duplicates in the dataset.

2.3.1 Handling 3-Fold Symmetry

Three-fold symmetry, also known as 3-fold rotational symmetry, is a type of rotational symmetry where a molecule looks the same after being rotated by 120 degrees (or 2π/3 radians) around a central axis. This means that if you rotate the molecule by 120 degrees, 240 degrees, or 360 degrees, its appearance remains unchanged.

To better understand three-fold symmetry, consider a molecule with three identical groups (such as Hydrogen, H) located at the vertices of an equilateral triangle. The symmetry axis passes through the center of the triangle and is perpendicular to its plane. When the molecule is rotated around this axis by

120 degrees, each Hydrogen moves to the position previously occupied by its neighbor. Because all three groups are identical, the molecule appears the same after the rotation.

Mathematically, the transformation of a torsion angle θ under three-fold symmetry can be described by the following expression:

$$\theta' = ((\theta - \pi/3)\, mod\, (2\pi/3)) - \pi/3$$

For specific values of θ:

- For $\theta = 0,\ \theta' = 0$
- For $\theta = 2\pi/3,\ \theta' = 0$
- For $\theta = -2\pi/3,\ \theta' = 0$

This means that when the torsion angle θ is either 0, 2π/3 or -2π/3, it remains unchanged under the three-fold rotational symmetry.

Chemical intuition identified three-fold symmetry in torsion angles t16, t17, t18, and t19. This symmetry makes these torsion angles effectively duplicates. Thus, a three-fold symmetry formula is applied to torsion angles t16, t17, t18, and t19.

2.3.2 Removing Duplicates Based on Torsion Values

After addressing the three-fold symmetry and normalizing the data, we now aim to identify and handle duplicates within our dataset. The objective is to apply an algorithm capable of detecting and managing duplicate or near-duplicate molecular conformations, thereby limiting subsequent computational costs for optimizing the structures at a higher level of theory. To achieve this, we will implement clustering algorithms that can group similar data points together, allowing us to identify and eliminate redundant entries effectively. This process will ensure that our dataset remains manageable and focused on unique molecular conformations, facilitating a more precise exploration of the underlying structural patterns.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is one of the most suitable and commonly used clustering algorithms. DBSCAN excels in detecting clusters of arbitrary shapes and sizes, which is particularly useful for identifying duplicates or near-duplicates in molecular datasets.

DBSCAN requires only two parameters: epsilon and min_samples.

- epsilon (eps): The maximum distance between two samples for them to be considered as in the same neighborhood.
- min_samples: The number of samples in a neighborhood for a point to be considered as a core point. This includes the point itself.

Setting min_samples to 1 makes the concept of noise points almost meaningless, as nearly every point will belong to some cluster unless it is not within ε distance of any other point. To ensure we retain non-duplicate points, we must consider noise.

After applying the DBSCAN algorithm with appropriate parameters and accounting for three-fold symmetry to the 1326 conformers provided by CREST, we have successfully reduced the dataset to 550 distinct clusters. From each cluster, we selected one conformer, resulting in a final dataset of 550 unique conformers for more precise exploration of molecular structures.

2.4 Processing and Analysis of 550 Conformers

After obtaining 550 conformers, we performed Density Functional Theory (DFT) calculations on these conformers using Gaussian. Subsequently, we utilized the PLUMED tool to extract the torsion angles for the 550 conformers. This extraction was followed by applying the same preprocessing steps as before, including addressing the three-fold symmetry in torsion angles t16, t17, t18, and t19. To further refine the dataset, we applied the DBSCAN clustering algorithm to the updated set of 550 conformers. By adjusting the clustering parameters, we managed to reduce the dataset to 244 distinct conformers.

For these 244 conformers, we took vibrational spectra across range of wavenumbers, from 0 to 4000 cm^{-1}. Below, two tables illustrate the updated dataset, showing how the conformers and their associated spectra are organized.

	Torsion 1	**Torsion 2**	**Torsion 3**	**…**	**Torsion 19**
Conformer 1					
Conformer 2					
Conformer 3					
….					
Conformer 244					

	Wavenumber 0	**Wavenumber 1**	**…**	**Wavenumber 4000**
Conformer 1	Intensity	Intensity	Intensity	Intensity
Conformer 2	Intensity	Intensity	Intensity	Intensity
Conformer 3	Intensity	Intensity	Intensity	Intensity
…	…	…	…	…
Conformer 244	Intensity	Intensity	Intensity	Intensity

2.5 Torsional and Spectral Differences

Finally, we computed the torsional and spectral differences among the 244 conformers resulting in a new dataset

1. The torsional difference dataset (Y) consists of 19 columns representing torsional differences and n^2 rows (where n = 244) representing different conformer pairs.
2. The spectral difference dataset (X) comprises 4001 columns representing spectral differences and n^2 rows (where n = 244) , representing different conformer pairs.

3. PROPOSED METHODOLOGY AND RESULTS

To achieve our goal of predicting torsional differences using spectral differences, we will employ regression algorithms. In this setup, our target variable (Y) is the torsional difference, and our predictor variable (X) is the spectral difference. To accurately model this relationship, we will apply two regression algorithms: Decision Tree Regression and Random Forest Regression.

Before applying these regression algorithms, we will use Principal Component Analysis (PCA) for dimensionality reduction of the spectral difference data (X). This step is essential to enhance the performance of our regression models.

3.1 Experimental Setup

In our experimental setup, we used PyTorch for data handling, Pandas and NumPy for data manipulation, Scikit-Learn for PCA, standard scaling, and model training with DecisionTreeRegressor and RandomForestRegressor, and Matplotlib for data visualization. We ran computations on Google Colab and utilized Graham Cedar with SLURM scripts for time-intensive tasks.

3.2 Dimensionality Reduction of Spectral Difference Data

Training machine learning models with vibrational spectra that span a wide range of wavenumbers, from 0 to 4000 cm^{-1}, presents significant challenges due to the high dimensionality of the data. Firstly, the computational complexity increases as more features demand more processing power and memory. Secondly, there is a heightened risk of overfitting, where the model might learn noise in the data rather than underlying patterns.

To address these challenges, Principal Component Analysis (PCA) is employed. PCA is a dimensionality reduction technique that transforms the original features into a new set of principal components, which capture the most variance in the data.

To ensure the quality of the data, Few steps were performed before PCA. Firstly, the spectral data was binned by summing intensity values over predefined bin sizes (in this case 10). A threshold was applied to set values below 5% of the maximum absolute value to zero, removing irrelevant points. Secondly, Columns with only zero values were removed to ensure that only meaningful features remained in the dataset. Then, the data was standardized to have a mean of zero and a standard deviation of one.

We retained components that explained 95% of the variance. These 49 principal components capture the majority of the information from the original 4001 spectral differences, reducing the data while preserving essential patterns and variations.

3.3 Training and Testing Dataset

The dataset was split into training and testing sets, ensuring that the testing set contained data points from one specific structure, typically the lowest energy structure, while the training set included all other points.

3.4 Decision Tree Regressor

The Decision Tree Regressor was employed to predict torsional differences from the PCA-transformed spectral data. This model splits the data into subsets based on feature values to minimize variance within each subset and forms a tree. Key parameters include:

- random_state: This is for shuffling the data. Setting it to “42”.
- max_depth: This is the maximum depth of the tree. When max_depth is None, the decision tree grows until every leaf node is a pure node. Limiting the depth helps prevent overfitting by controlling the model's complexity.

Considering default parameters, the results are:

Torsions	Train MSE	Test MSE
Torsion 1	0.00	0.000000280
Torsion 2	0.00	0.010870718
Torsion 3	0.00	0.001486611
Torsion 4	0.00	0.000121346
Torsion 5	0.00	0.000000513
Torsion 6	0.00	0.000003377
Torsion 7	0.00	0.001032787

Torsion 8	0.00	0.000603395
Torsion 9	0.00	0.000006643
Torsion 10	0.00	0.000371076
Torsion 11	0.00	0.094639709
Torsion 12	0.00	0.625024238
Torsion 13	0.00	0.611093431
Torsion 14	0.00	0.645655989
Torsion 15	0.00	0.539478432
Torsion 16	0.00	0.001229962
Torsion 17	0.00	0.000031403
Torsion 18	0.00	0.000010703
Torsion 19	0.00	0.000024509

The train MSE for all torsions is 0.00, indicating perfect fitting on the training data. However, this may suggest overfitting. The test MSE varies significantly across different torsions. Some torsions (e.g., Torsion 1, Torsion 5) have very low test MSE, indicating good generalization. Others (e.g., Torsion 12, Torsion 13, Torsion 14, Torsion 15) have much higher test MSE, suggesting poor generalization and potential overfitting.

We need to hyper-tune the parameters like max_depth, splitter, and max_features for Torsions 12, 13, 14, and 15, as they have higher test MSE, which is not good and may be due to overfitting. These adjustments will help to reduce overfitting and improve the model's performance on these torsions.

These are the updated results after hypertuning of parameters:

Torsions	**Train MSE**	**Test MSE**
Torsion 12	0.41624270	0.38784776
Torsion 13	0.29393417	0.31195776

Torsion 14	0.29311718	0.33268522
Torsion 15	0.29032123	0.23975463

3.5 Random Forest Regressor

A Random Forest Regressor is a machine learning algorithm that builds a collection of decision trees during training. Each tree in the forest predicts the output, and the final prediction is the average of individual tree predictions.

By averaging the predictions of multiple trees, Random Forests reduce the variance of the model, leading to more robust and reliable predictions.

Considering default parameters, the results are:

Torsions	**Train MSE**	**Test MSE**
Torsion 1	0.000000600	0.000004367
Torsion 2	0.000000537	0.005769001
Torsion 3	0.000000162	0.000801623
Torsion 4	0.000000535	0.000035855
Torsion 5	0.000000401	0.000000481
Torsion 6	0.000001056	0.000001228
Torsion 7	0.000005856	0.000292320
Torsion 8	0.000003604	0.000121210
Torsion 9	0.000000125	0.000000827
Torsion 10	0.000001646	0.000128965
Torsion 11	0.000032964	0.067277609
Torsion 12	0.020890856	0.289524127
Torsion 13	0.015080970	0.233034084
Torsion 14	0.027842688	0.332789987

Torsion 15	0.028862975	0.233445433
Torsion 16	0.000077196	0.000293705
Torsion 17	0.000002675	0.000008811
Torsion 18	0.000000983	0.000003436
Torsion 19	0.000000818	0.000014584

4. CONCLUSION

This study demonstrates the potential of machine learning in accelerating and enhancing molecular structure elucidation through vibrational spectroscopy. We employed Decision Tree and Random Forest Regressors, reducing spectral data dimensionality with Principal Component Analysis (PCA) to enhance model performance. Initially, the Decision Tree Regressor showed perfect fitting on training data but significant test MSE variation, indicating overfitting, with an average test MSE of 0.1332. After hyperparameter tuning, the average test MSE improved to 0.0728. The Random Forest Regressor, averaging multiple decision trees' results, demonstrated superior performance with an average test MSE of 0.0612, indicating robust and reliable predictions.

5. FUTURE WORK

For future work, We plan to experiment with smaller conformer sets of THC, rather than utilizing the full dataset, to analyze the results and determine if smaller subsets can yield comparable insights. Additionally, we will extend our methodology to other molecules beyond THC to evaluate the generalizability of our approach across different chemical structures. Implementing and evaluating other regression models will be a priority to identify models that offer better performance. Besides Mean Squared Error (MSE), we will also consider other evaluation metrics such as Root Mean Squared Error (RMSE), R-squared (R^2) score, and Mean Absolute Error (MAE). Furthermore, hyperparameter tuning for all regression models will be conducted to optimize their performance and reduce overfitting. We also aim to convert the regression problem into a classification problem and train classification models, analyzing the classification results to see if this approach provides any advantages in terms of accuracy and computational efficiency. By addressing these aspects, we aim to further enhance the utility and effectiveness of machine learning in the realm of vibrational spectroscopy and molecular structure elucidation.