Graph Neural Networks For Learning Molecular Excitation Spectra

Kanishka Singh,[†,‖] Jannes Münchmeyer,[‡,¶] Leon Weber,[¶,§] Ulf Leser,[¶] and Annika Bande[*,†]

†*Helmholtz-Zentrum Berlin für Materialien und Energie GmbH,Hahn-Meitner-Platz 1, 10409 Berlin, Germany*

‡*Deutsches GeoForschungsZentrum GFZ, Telegrafenberg, 14473 Potsdam*

¶*Humboldt-Universität zu Berlin, Unter den Linden 6, 10117 Berlin, Germany*

§*Max Delbrück Center for Molecular Medicine in the Helmholtz Association, Robert-Rössle-Straße 10,Berlin 13125, Germany*

‖*Institute of Chemistry and Biochemistry, Freie Universität Berlin, Arnimallee 22, 14195 Berlin, Germany*

E-mail: annika.bande@helmholtz-berlin.de

Abstract

Machine learning (ML) approaches have demonstrated the ability to predict molecular spectra at a fraction of the computational cost of traditional theoretical chemistry methods while maintaining high accuracy. Graph neural networks (GNN) are particularly promising in this regard, but different types of GNNs have not yet been systematically compared. In this work we benchmark, and analyze five different GNNs for the prediction of excitation spectra from the QM9 data set of organic molecules. We compare GNN performance in the obvious runtime measurements, prediction accuracy, and analysis of outliers in the test set. Moreover through t-map clustering

and statistical analysis, we are able to highlight clear hotspots of high prediction errors as well as optimal spectra prediction for molecules with certain functional groups. This in-depth benchmarking and subsequent analysis protocol lays down a recipe for comparing different ML methods and evaluating dataset quality.

Introduction

In recent years, various approaches rooted in machine learning (ML) have demonstrated the ability to tremendously speed up the chemical characterization of molecules. In a nutshell, they learn patterns in the structure of molecules with known properties and use these patterns to predict the same properties of novel materials. Compared to the traditional quantum-chemical methods, they are simple to implement and perform predictions much faster while maintaining chemical accuracy in their predictions.[1]

The performance of ML methods depends on the particular property to be predicted.[2,3] Chemical properties are associated either to ground or to excited states of molecules. A molecule in a ground state is in its most stable form and thus at its lowest energy. Interactions with light or matter transmit energy to it, which causes excitations of electrons or bond vibration, bringing the molecule into its excited states. ML methods predominantly have been applied to study ground state properties, such as solubility, ground state energy, or dipole moment.[4] Properties of excited molecules are more challenging to compute as they involve intricate changes of the electronic structure within the molecule itself.

The spectrum of a molecule that interacts with light is a particularly important excited-state property. Spectroscopy has been one of the most crucial ways of identifying the composition of unknown materials since centuries,[5,6] and it still is. Several techniques for experimentally measuring a molecule's spectrum exist, but they often need to be supported by calculated spectra that use theoretical chemistry methods. Methods from time-dependent density-functional theory (TD-DFT) can be used to obtain such a spectrum *in silico*, but its runtime explodes when a larger number of excited states must be included in the com-

putation.[7] With the growing pace at which new materials are required to satisfy human demands, new, faster, and at least equally accurate methods are necessary to obtain spectra computationally, for instance to identify promising materials before their actual synthesis.[8]

Early efforts for the ML-based prediction of molecular spectra have been presented in,[9] whereby TDDFT-calculated UV spectra were used to train machine learning models. A cornerstone work[10] showed that the usage of deep tensor neural networks (DTNN)[11] leads to new state-of-the-art results for the prediction of spectra of frequently-used QM9 data set of organic molecules. DTNNs are graph neural networks (GNNs) that represent the molecules under study with matrices representing charges and distances. Several other GNNs have been used to predict molecular properties,[12,13] infrared spectra,[14–16]photoemission spectra[17] or chemical shifts in nuclear-magnetic resonance spectra.[18] However, there is to date no in-depth study comparing different types of GNNs in their performance for predicting spectra. On top, a detailed analysis of the strength and weaknesses of the different methods with respect to the prediction of chemical properties of the molecules under study is still missing.

In this study, we benchmark five different recently proposed GNNs regarding their ability to accurately predict spectra. We discuss first the different message passing protocol and different GNN architectures that are evaluated in this work, namely graph convolution networks (GCN),[19] the message passing neural network (MPNN),[1] SchNet,[20] the graph isomorphism network (GIN),[21] and the deep tensor neural networks (DTNN).[11] Using the QM9 set of more than 130k molecules as benchmark set, we measure and compare the GNNs' prediction accuracy and runtime to assess the feasibility of applying such prediction methods over truly large data sets. We also provide in-depth analysis of the performance of these methods for different classes of molecules and their functional groups to identify their mutual strengths and weaknesses. This contribution is laying down a systematic framework that evaluates different models on their accuracy as well as their prediction quality in conjunction with different chemical features of the molecules.

Methods

Representing molecules as graphs

Employing ML for predicting a molecule's property requires to first represent its structure in a suitable form. Two typical approaches are lists of either exact Cartesian coordinates or of relative coordinates of each atom in the molecule. However, this does neither acknowledge for the bonding between atoms nor can it represent the nature of these bonds. SMILES (simplified molecular input line entry specification)[22] encodes such information but disregards geometry. Coulomb matrices model the interaction of each atom with the others, and thus indirectly encode spatial information, but they cannot encode bonding information.[23] A rather natural way of representing molecules are mathematical graphs, where nodes in the graph represent atoms in the molecule and edges represent bonds between atoms. By attaching structured labels to nodes/edges, respectively, graphs can represent arbitrary atom/bond properties, such as distances or angles. A particular attractive feature of the graph representation is that it provides translational and rotational invariance.[13]

Formally, a molecular graph is a structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes, each representing one atom, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges, each representing one bond between atoms. Edges may be defined either through pre-existing knowledge about molecular bonds or by connecting atoms that lie within a certain physical distance to each other. Properties of an atom $v \in \mathcal{V}$, such as atomic number or hybridization in a molecule, are modeled as an f-dimensional node feature vector $\mathbf{x}_v \in \mathrm{R}^f$, properties of an edge $v \times w \in \mathcal{E}$ are modeled as a d-dimensional edge feature vector $\mathbf{e}_{vw} \in \mathrm{R}^d$. When attributes of atoms or edges have categorical values (such as being 'aromatic' or 'non-aromatic'), they typically are transformed into numbers using one-hot encodings.[24]

Figure 1 demonstrates how the molecule acetic acid is represented as a molecular graph. Panel **1** shows the molecular structure as skeletal formula, a standard chemistry representation, whereas Panel **2** depicts its graph representation, with atoms replaced by an appropriate

node symbol and bonds replaced by an edge symbol. Panel **3** gives the feature vector of nodes (edges) of a certain symbol (Bond lengths are measured in Å).

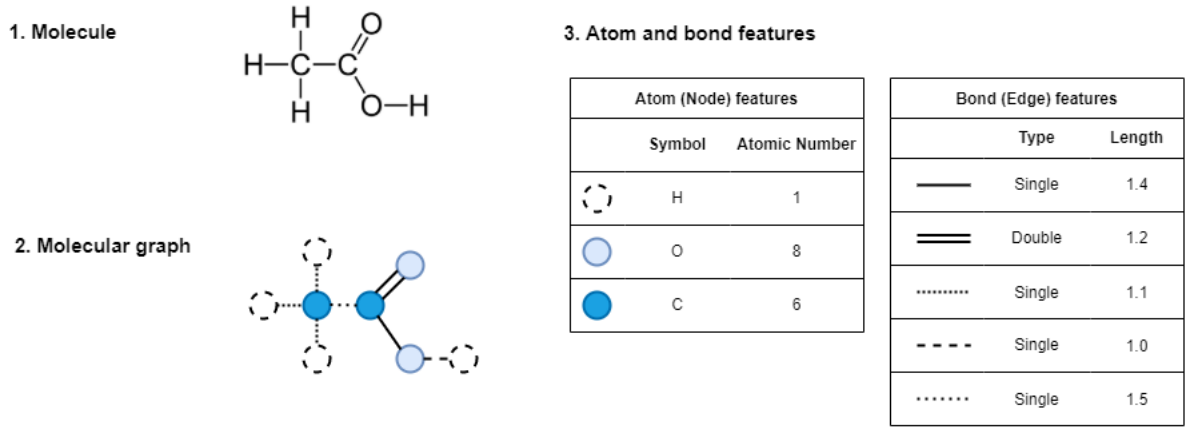


Figure 1: Correspondence between a molecular structure (1) and its graph representation (2), as well as features describing the atoms and edges of the graph (3).

Graph Neural Networks

GNNs are a specific type of artificial neural networks (ANNs) that operate on graphs. In a nutshell, ANNs operate on the feature representation of an object and consist of a sequence of simple differentiable and parametrized transformations which, when combined appropriately, allow to model arbitrarily complex functions and produce some output, typically in the form of a vector. The individual transformations are typically arranged in layers. The output x^l of the l^{th} of these functions f^l is called representation at layer l, with x^0 denoting the input of the ANN. Accordingly, the transformation performed by l can be written as $x^l = f^l(x^{l-1})$. For vanilla (feed-forward) ANNs, x^l is a tensor, i.e., a multi-dimensional vector, of constant size. ANNs are trained by fitting their parameters to a set of training data, consisting of pairs of input features with the desired output vector. A training procedure determines optimal parameters of the individual transformation functions by performing some form of gradient descent, an optimization method that makes use of the differentiability of the functions f^l. By setting aside a part of the dataset for testing, the performance of the trained model can

be assessed on instances unknown to the model. More often than not, the dataset is also split into another part, called the validation set, which is used to prevent over-fitting of the model and keep track of variables during training.

In contrast, GNNs arrange layers according to a graph structure. Therein, they represent nodes (edges) as node (edge) embeddings, which are vectors of arbitrary dimensionality created from the node (edge) feature vectors $\mathbf{x_v^l}$ ($\mathbf{e_{v,w}^l}$) by either a lookup operation into an embedding matrix or by applying a simple feed forward network. The transformation functions f^l are designed such that they take the underlying graph structure into account. Training typically is performed by a message passing mechanism in which multiple rounds of differentiable message passing between neighboring nodes allow them to exchange information leading to updates of their embeddings. The specific instantiation of a message passing GNN can be described by its choices for its three basic operations, namely AGGREGATE, UPDATE, and READOUT. The first two specify in which way nodes exchange information in a cycle, while the latter collects the information that is spread across the node representations into a single global representation for the whole graph. This cycle of exchanging and updating node information is collectively termed as a message passing cycle as depicted in **Figure 2 (c)**. The number of message passing cycles (layers) controls the amount of information that can be gained during the training process, and is a hyper-parameter that depends on the GNN being employed. Gilmer et al.[1] surveyed existing message-passing GNN approaches for the prediction of molecular properties in the QM9 data set such as dipole moment, band gap, and free energy (but not spectra), and analyzed the performance with different choices for AGGREGATE, UPDATE and READOUT. We describe the three base operations mathematically below, as well as pictorially through **Figure 2 (b)**.

AGGREGATE: The AGGREGATE operation is executed for every node in the graph once per layer. For a node v, it aggregates localized information from the embeddings of its neighbors into an intermediate vector $\mathbf{m}$. In **Figure 2, this step occurs in the yellow**

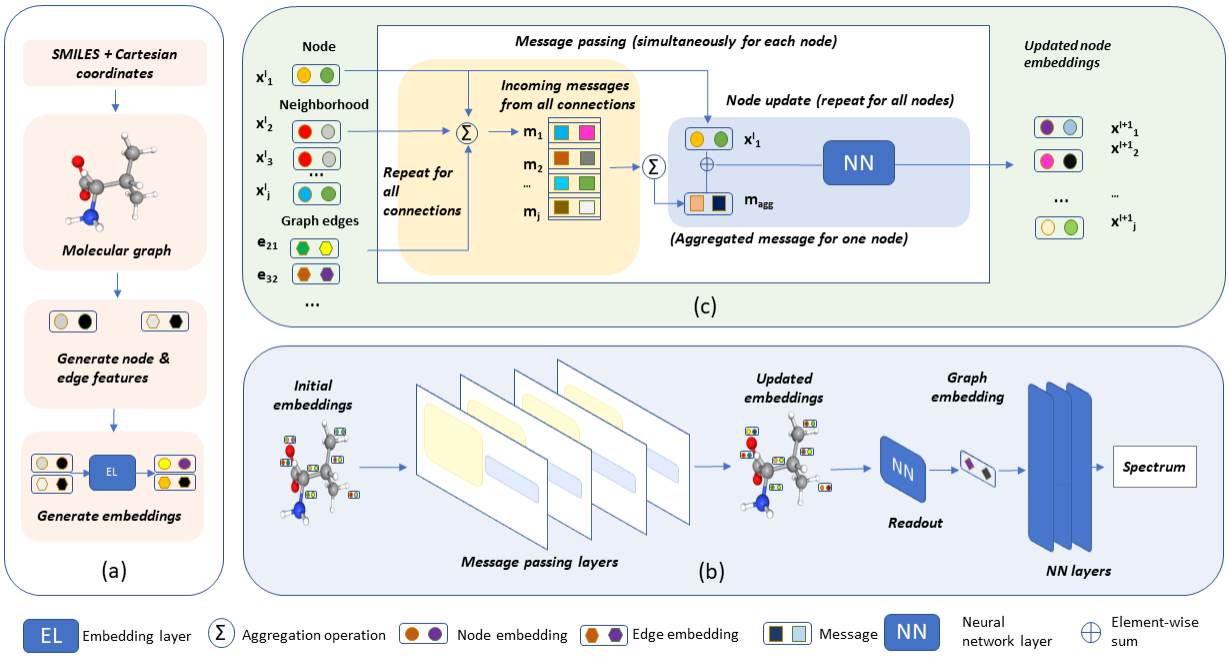


Figure 2: A typical GNN model used in this study needs as an input a molecular structure converted into a graph with embeddings(a). Molecular graphs from the dataset undergo several message passing cycles to generate graph level embeddings(b) for which one message passing layer is detailed in (c).

regions of each message passing layer. This step is formally described as

$$\mathbf{m}_v^{l+1} = \mathbf{AGGREGATE}^{l+1}\left(\left(\mathbf{x}_w^l, \mathbf{x}_v^l, \mathbf{e}_{v,w}^l\right)_{w \ \epsilon \ \mathcal{N}(v)}\right) \tag{1}$$

where $\left(..\right)$ denotes a multiset and $\mathcal{N}(v) \subseteq \mathcal{V}$ defines the neighborhood of a node v in the graph G.

UPDATE: For every node v, the UPDATE step uses its aggregated localized information m and the current node embedding $\mathbf{x}^l$ to calculate the next embedding $\mathbf{x}^{l+1}$. This step occurs in the blue part of the each message passing layer in **Figure 2**. It is formally defined as

$$\mathbf{x}_v^{l+1} = \mathbf{UPDATE}^{l+1}(\mathbf{x}_v^l, \mathbf{m}_v^{l+1}). \tag{2}$$

Notably, both AGGREGATE and UPDATE are applied to each node individually.

READOUT: After having computed the new embeddings for all L layers, all node representations are combined in the READOUT step, which is formally defined as

$$\mathbf{x}_g = \mathbf{READOUT}(\mathbf{x}_v^L \mid v \, \epsilon \, G). \tag{3}$$

The READOUT phase produces a vector representation of the graph, which can then be used as an input for the prediction of spectra by feeding it to the output layer. This is the final step of the message passing framework, and is labelled Readout in **Figure 2**.

In this work, we compare five concrete GNNs regarding their ability to accurately and quickly predict the excitation spectrum of a molecule represented as a graph, namely SchNet,[20] MPNNs,[1] GCNs,[19] DTNN[20] and GINs.[21] MPNN models use a special readout function (set2set[25]), which provide them an advantage over other readout functions such as averaging or summation. GINs have been designed specifically with the purpose of overcoming deficiencies in GCN and MPNN that fail to distinguish between certain molecular graphs. GINs are the only of the five GNNs for which we employ edge embeddings. SchNet and DTNN frameworks share a common architecture; with SchNet improving upon the ability to capture subtle changes in atomic position compared to other frameworks. In contrast to the other four GNNs, DTNN operates on two matrices, one representing the distances between atoms, and a charge vector denoting nuclear charges. SchNet does not use explicit edge/bond representations but infers edges from spatial vicinity of the atoms using a threshold. A detailed explanation of the mathematical differences of these GNNs can be found in the supplement. We discuss GNNs that only update the node embeddings of the molecular graph. GNN frameworks where edge embeddings are also updated[26] have also been employed to predict molecular properties with great success. Message passing models for GNNs in chemical application are rapidly progressing. The most recent development lies in equivariant message-passing models such as PAINN (Polarizable Atom Interaction Neural Network).[27] Since molecules are three dimensional entities, conventional message passing algorithms are

not able to fully capture such directional information. Equivariant message passing enables creation of rotationally invariant representations of molecules which solve this issue and are able to distinguish between structures that differ only in their three dimensional orientation in space such as conformers. Therefore, they were found to be better than existing methods in predicting several molecular properties. These models could probably also improve upon the results of this study.

Dataset and hyper-parameter optimization

Dataset

We perform our analyses on the QM9 data set[28] extended with spectral data.[10] This data set, which we term QM9* for the purpose of this study, contains 132,531 organic molecules composed of the first- and second row main group elements H, C, N, O, and F. The property we study is a spectrum taken from Ref.,[10] which the authors computed from the 16 lowest Kohn-Sham orbital energy eigenvalues from a DFT calculation on the QM9 molecules with Gaussian line broadening. Although this spectrum has limited practical applicability, because it cannot be obtained from measurements, it is an ideal candidate for the systematic comparison of ML techniques for spectroscopy applications, because its computation is much less time consuming than that of TD-DFT spectra that directly compare to experiment. This way a dataset of this size was at all achievable.

The spectra in the data set are available as X, Y tuples, with the X coordinate discretized into 300 points between -30 and 0 eV, and the Y coordinate giving intensity values ranging from 0 to 5 units. The original data set contained a few very small molecules, such as CH_4, for which some of the lowest-energy excitations lie far below -30 eV (in the range around -200 eV).[10] We consider 12 such small molecules as outliers and clean the data set by removing these molecules, leaving 132,519 molecules for our evaluation.

For training, optimization, and evaluation, the data set is split into a into test, training,

and validation set using the **90-5-5** ratio as provided by Ghosh et al.[10]. **Table 1** lists the node and edge features that are likewise used in the MPNN, GCN, SchNet and GIN models; DTNN does not use such features. For the SchNet model, neighbors are inferred implicitly using physical distances of atomic coordinates. Molecular features such as atomic number, donor-acceptor properties, hybridization of constituent atoms etc. were extracted using the RDKit library[29] from SMILES representations of the molecule while bond distances were extracted from xyz coordinates, both of which were provided in the dataset.

Table 1: Characteristics of atoms and bonds (node and edges) as represented in the node feature vector in conjunction with their respective type if encoding

Node feature	**Encoding**
Atomic number	One hot
Acceptor	One hot
Donor	One hot
Hybridisation	One hot
Aromaticity	One hot
Number of hydrogens	Integer
Edge feature	**Encoding**
Bond distance	Real
Bond type	One hot

GNN models and hyper-parameters

Give the excessive compute times required by DTNN (see **Table 3** below), we did not recompute its results on the QM9* data set but report values from the original publication.[10] Note that this could introduce a small error due to the differences in data sets after removing the 12 outliers for all the present calculations. However, all these 12 molecules are part of the training split, which implies that results on the test split are fully comparable. The other four models we evaluate are trained using the pytorch-geometric library.[30] For the GIN and GCN model, our implementations builds on the work of Hu et al.[31], the implementations for which can be found at.[32] For the SchNet and MPNN models we use and adapt the codes as provided in the pytorch-geometric library itself.[30] All models were modified to produce

output vectors of length 300 representing the predicted spectrum of the molecule. All models were trained for 300 epochs. Models are trained using the ADAM optimizer,[33] RMSE (root mean square error, discussed in the next section) as loss function, and an initial learning rate of 0.01, which is reduced by a factor of 0.9 each time the validation loss plateaus after 5 epochs. Relative Spectra Error(RSE, discussed in the next section) and RMSE loss are both monitored on the validation set for every 50 epochs (at which each model is saved) in order to check for over-fitting. We select the model that has the best RSE and validation loss for each GNN. Representative learning curves for different GNNs can be found in the supplement. Training was performed on an Nvidia Tesla P100 16GB GPU.

Hyper-parameter selection was performed using a grid search over four variables: number of message passing cycles, atomic level embedding dimension, graph level embedding, and batch size. Other model-specific variables were fixed as in the original implementations:[30] the number of filters in SchNet at 200, and the cutoff radius for defining the atom neighborhood at 10 Å. In total 300 models were trained for each of GIN, GCN and SchNet, and 180 models were trained for MPNN (since the graph level embedding is not explicitly a part of the training parameters in the model, but dependent on the atomic level embedding. The entire range of parameters for tuning can be found in Tables S1 and S2 of the supplement). The resulting hyper-parameters are reported in **Table 2**.

Quantifying the error in predicted spectra

Predicting molecular spectra essentially is a regression problem. RMSE and MAE (mean absolute error) are the most commonly used formulations to quantify the prediction errors of a ML model in such problems. However, as spectra are series of data points along the energy levels, we follow[10] and use the RSE for comparing different GNNs. Intuitively, the RSE is a normalized version of the RMSE, obtained by dividing the RMSE by the total spectral energy of the target. Formally, it is defined as follows. Let $\mathbf{y^{tar}}$ and $\mathbf{y^{pred}}$ denote the vectors for the target and the predicted intensities of the signal at an energy E. Let dE

Table 2: Optimal hyper-parameters for the four GNNs we trained in this study as obtained from a hyperparameter grid-search. The last row gives the number of trainable parameters per GNN

Hyper-parameter	**GCN**	**GIN**	**SchNet**	**MPNN**
Node embedding	40	40	100	150
Edge embedding	40	40	NA	NA
Message passing cycles	5	5	6	3
Set2Set steps	NA	NA	NA	5
Batchsize	50	50	50	200
Cutoff radius	NA	NA	10	NA
Num. of filters	NA	NA	200	NA
Parameters	0.74mn	2.09mn	1.77mn	4.01mn

be an infinitesimal difference in energy across the energy axis and $E_{max} - E_{min}$ denote the range of energies under consideration. The RSE is then defined as

$$RSE = \frac{\sqrt{\int (y^{tar} - y^{pred})^2 dE}}{\int y^{tar} dE}. \tag{4}$$

For approximating this term from a discrete spectrum with N points we defined ΔE, the unit distance between two points as:

$$\Delta E = \frac{E_{max} - E_{min}}{N}. \tag{5}$$

With this in place, the RSE is approximated as

$$RSE = \frac{\sqrt{\sum_i^N (y_i^{tar} - y_i^{pred})^2 \cdot \Delta E}}{\sum_i^N y_i^{tar} \cdot \Delta E}. \tag{6}$$

A small relative spectral error means that the predicted spectrum is a good reproduction of the original spectrum. **Figure 3** shows two comparison between target spectra (blue solid line) and their ML-predicted counterpart (orange, dashed line): one with a small error (RSE=0.009,a)), one with a large error (RSE=0.208,b)). For a) the two lines coincide almost

perfectly, with small differences at the two peak maxima ($\leq 5\%$). Here, RSE and RMSE produce a very similar value close to zero. In contrast, differences are large in the spectra of b). The four clear peaks of the target spectrum at energies about -16, -12, -7, and -5 eV are also roughly predicted by the ML method, but intensities deviate considerably. In addition, a few further peaks are predicted at high energies where the target spectrum actually has zero intensity. Such strong deviations result in an unbound RMSE between the two spectra, while the normalized RSE remains within the $0-1$ range, which enhances interpretability.

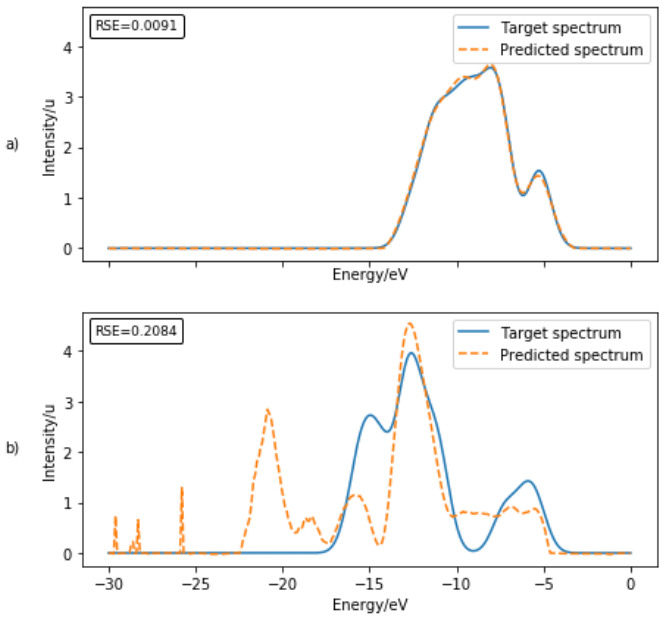


Figure 3: Two plots of a target spectrum computed from Kohn-Sham eigenvalues (blue, solid line) and a corresponding spectrum predicted with GIN (orange, dashed line). The absorption intensity is plotted on the abscissa against the energy on the ordinate.

Results

We divide the presentation of results in two sections. We first discuss the accuracy achieved by the five GNNs over the entire data set. We present average and extreme RSE values, show distribution of errors, and also consider the time it took to train the models. Given

that the current state-of-the-art was achieved by the DTNN in 2019, this comparison will help to show the progress that has been made since then.

The prediction quality of a model does, however, not only depend on the technology being used, but likewise on the particular sets used for training and test. Therefore, in the second part of this chapter we present insights on the properties of a molecule which affect the quality of predictions, focusing on molecular weight, size of molecules and structural features such as common functional groups. This will help to identify groups of molecules whose spectra are particularly easy or particularly hard to predict, to relate these differences to properties of the prediction model, and to lay the basis for a systematic improvement of models regarding the difficult cases.

GNN impact on the relative spectral errors

As a measure of the overall performance of the various models, **Table 3** summarizes their average RSE over all 6627 molecules from the test set as well as the number of epochs used during training.

On an average the SchNet outperforms all other GNNs, with a 20% improvement on the average RSE of the previous DTNN results. It is better in all other metrics as well except the maximal RSE, where DTNN reaches a better value. GCN performs poorly, DTNN, GIN and MPNN perform similar to each other. The inferior performance of GCN possibly roots in its much lower number of trainable parameters (see **Table 2**). The four more recent GNNs need orders of magnitude less training time than DTNN due to their much lower number of needed epochs.

The minimal and maximal RSE values in **Table 3** represent the lowest and highest RSE value for a molecule in the test set, respectively. The minimal RSE of SchNet, GIN and DTNN are comparable, while GCN and MPNN never achieve such good fits. The maximal RSE value for the DTNN model is the lowest of all models, indicating that the DTNN manages to achieve lower maximal error margins, possibly due to the extremely high number

Table 3: Prediction accuracy of different GNN models. Results for DTNN are taken from.[10] SchNet outperforms all other methods in almost all metrics. Bold-face numbers indicate the optimal value of the metric across the GNNs

Metric	**GCN**	**GIN**	**SchNet**	**MPNN**	**DTNN**
Epochs	300	300	300	300	10,000
Training times	5.5 hrs	7.5 hrs	4.5 hrs	6.0 hrs	13 days
Average RSE	0.039	0.029	**0.023**	0.041	0.029
Min RSE	0.011	0.005	**0.005**	0.01	0.006
1st Percentile	0.017	0.010	**0.009**	0.017	0.015
25th Percentile	0.029	0.020	**0.016**	0.030	0.027
50th Percentile	0.036	0.026	**0.021**	0.038	0.034
75th Percentile	0.045	0.035	**0.027**	0.048	0.043
99th Percentile	0.084	0.080	**0.060**	0.094	0.075
Max RSE	0.22	0.253	0.205	0.233	**0.135**

of training epochs - at the cost of extremely long training times. We did not test whether this value would also improve for the other models for much larger numbers of training epochs. To further understand the distribution of RSE values, we also include in the table percentile values at symmetrical intervals. For instance, the 99th percentile of SchNet is considerably lower than that of DTNN while its maximal RSE is higher, meaning that for almost all cases SchNet achieves better results but performs worse within the last 1% of molecules.

Differences between DTNN (as current state-of-the-art) and SchNet (as best method according to our benchmark) are further highlighted through a histogram of RSE values in **Figure 4** (similar plots for the other models can be found in the supplementary material, figures S5-S7). For instance, 98% of all molecules are predicted by SchNet with an error of at most 0.05, while this holds for only 87% of molecules for DTNN.

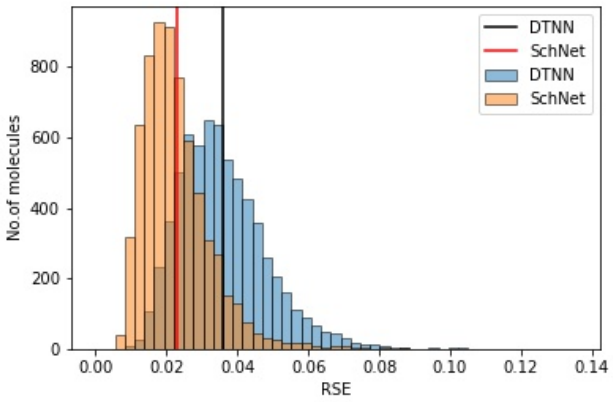


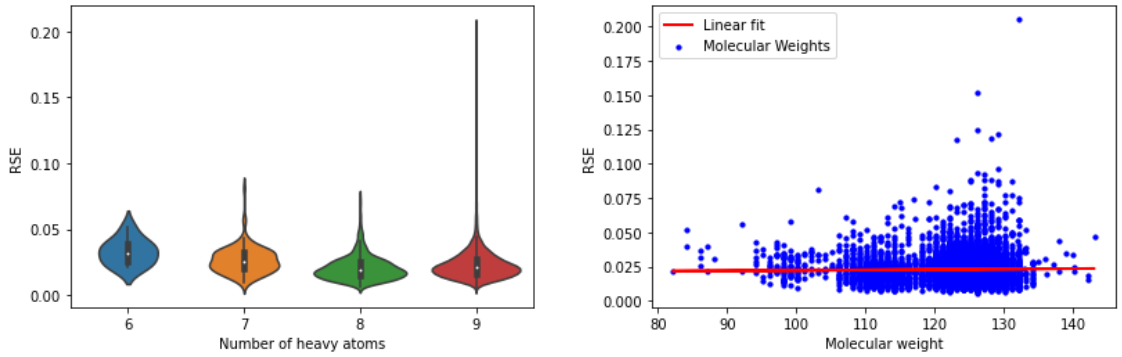
Figure 4: Histogram of the RSE values for the predicted spectra for SchNet (orange bars) and for DTNN (blue bars). The red and black line mark the respective average RSE.

Data set impact on the relative spectral errors

A general performance analysis as shown in the previous section does not answer the question of how well a model performs for different types of molecules. In this section, we therefore provide an analysis of how the RSE of spectra from the test set varies with molecular properties such as the number of atoms, molecular weight, and occurrence of certain important substructures. As a final step in the analysis, a high level visualization of RSE values across molecules of the test set is presented based on structural similarity metrics. For all the analyses carried out, the best performing models of each GNN were used to predict the spectra for the test set.

RSE by number of heavy atoms. **Figure 5a** illustrates the distribution of the RSEs for SchNet as function of the number of heavy atoms (i.e. second-row atoms of the periodic table) in the molecules of the test set. For molecules with six heavy atoms (leftmost plot), RSE values range from 0.02 to 0.05 with a roughly normal distribution slightly shifted to lower values. For molecules with more heavy atoms, the range of values increases, whereas the mode of the distribution decreases. From 0.04 for six heavy atoms to 0.03 for nine heavy atoms. Furthermore, the distribution becomes more narrow for more heavy atoms, meaning that more molecules have RSE close to the mode. This, however, can be attributed

to the simple fact that the number of molecules increases as we go from six to nine atoms. There are only very few RSE values beyond 0.10, in which cases the ML spectrum can be considered a poor prediction. Such values only occur for molecules with nine heavy atoms. Here, the largest RSE is 0.20, while for the molecules with eight atoms the highest RSE is 0.07. Violin plots for the four other models can be found in the supplementary material (Figures S8-S10). Comparing them with the SchNet plot again underpins the quality of SchNet.



(a) (b)

Figure 5: RSE variations with number of heavy atoms(a) and molecular weight(b) for SchNet

RSE by molecular weight. **Figure 5b** illustrates the variation of RSE values from SchNet with the molecular weight of molecules. As expected from the average metrics, most values cluster around 0.02, with only few outliers above 0.10. We find no correlation between the molecular weight of the molecules in the test set and the RSE values, as one would expect, and as seen in the figure with the slope of the line being of the order 10^{-5}. Similar plots for other GNNs can be found from Figures S11-S13.

Correlation of RSE with specific substructures. Recently it was shown that groups of molecules sharing certain functional groups, such as flouride or cyclo-alkanes, lead to better/worse performance in ML-based property prediction.[34] To test this observation with different types of GNNs, we divided the test set into groups based on the molecular functional groups by matching their corresponding substring in the SMARTS[35] representation of the molecules. To this end we first prepared a non-exhaustive list of functional groups and

calculated their frequency of occurrence in the test and training datasets. As shown in **Table 4**, the training and test occurrences of most functional groups deviate only by $0.05 - 3$ % from each other. This demonstrates a uniform and unbiased sampling in the data. However, not all functional groups occur equally often: Some groups are very frequent, such as ethers which are contained in 44% of the structures, while others are rare, like for instance carboxylic acids with a prevalence of only 1%. The complete list of functional groups can be found in table S3 of the supplement.

To understand the effect of functional groups across each model, we computed for each group and each GNN the average, minimal, and maximal RSE value. We next computed the significance of the deviation of the group-wise average RSE from the data set RSE excluding these molecules (two sided t-test) and sorted groups according to the resulting p-value. Intuitively, this p-value gives the probability that a deviation in average RSE as observed (or larger) between a subgroup and all other molecules occurs just by chance.

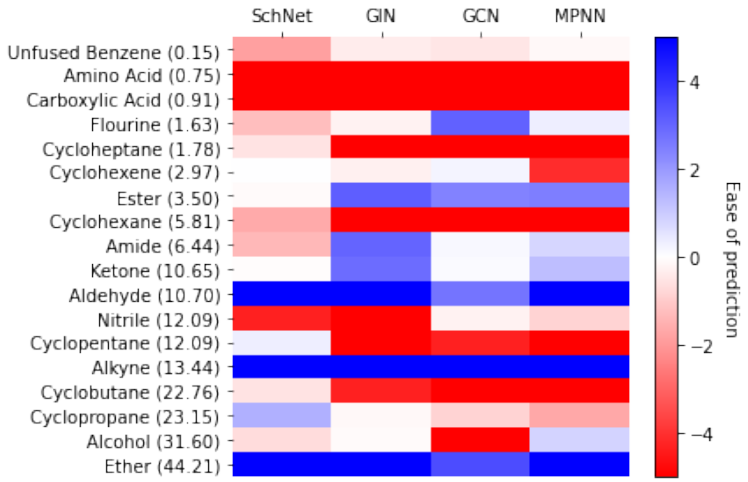


Figure 6: Ease of prediction across different functional groups for the four models. Across the vertical axis, functional groups are represented along with their percentage occurrence in the test set. Colors indicate if the performance of the said group is better (greater than 0 /blue) or worse (less than 0/red), compared to mean performance.

The heat map in **Figure 6** results from performing the p-value evaluations for all four models. Here, we evaluate 'Ease of prediction' (For an explanation of this parameter we direct the reader to the supplement) for each functional group, which depends on whether

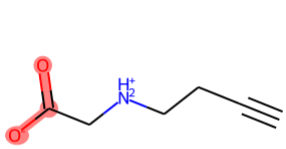
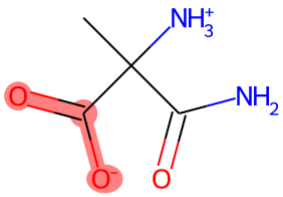
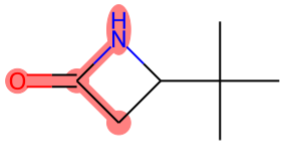
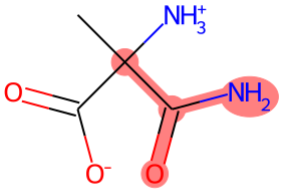
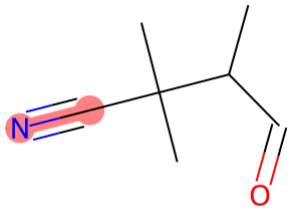
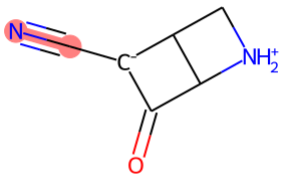
the mean of the RSE of molecules containing a given functional group is better than or worse than for the other molecules. The ease of prediction reflects how the prediction compares to the average performance of the model only. Thus, across the four models for fluorine the GCN is has a higher 'ease' implying that relative to other groups in the dataset, the GCN predicts other functional groups better than it predicts molecules with fluorine functional. This could still mean that the for a relatively easy group the for GCN the value of average RSE itself, could be higher than the equivalent for SchNet. Further, the figure shows that molecules with amino acid and carboxylic acid groups are hard to predict for all models, compared to other functional groups which hints to an intrinsic property of such molecules. In contrast, the spectra of molecules with ether or aldehyde groups are easy to predict. An interesting group is amide, where SchNet, the top performing model overall, achieves a performance worse than the three other GNNs. Another interesting observation is that SchNet performs better at predicting spectra for cyclic structures compared to other models. The analysis thus highlights the differences in prediction abilities for different functional groups across the models trained.

Table 4 shows the average test set RSE values for the most difficult or most simple functional groups. Along with this, the individual structures with the lowest and highest RSE values for each functional group are listed. Note that the difference is in the highest and lowest RSE values per group are often rather large. It is only in the case of carboxylic acids and amino groups that the average RSE is more than twice of the average RSE for the entire test set. Both these types of functional groups have charged entities, and often occur together as amino acid molecules in the data set. While these results shed light on functionals that impact the analytic power of GNNs in general, it is important to also consider that molecules are composed of several functional groups (for example the molecule with lowest RSE value for 'Carboxylic acid' also occurs in the 'Amide' functional results). A complete version of the table can be found in Table S4 of the supplement.

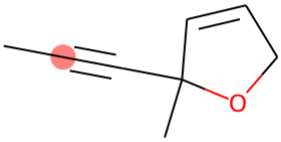
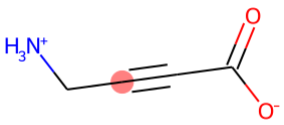
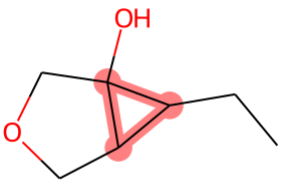
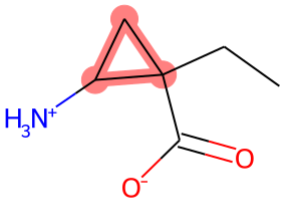
As a final observation, it is noted that for almost all structures with high RSE, that is

poorly performing molecules, atoms with charge are present in their structure. However, the two groups that contain charged entities, i.e., the amino and carboxylic acid group, form about only 2% of the training set, which could be a reason for this poor performance.

Table 4: Summary of RSE values and occurrences of different functional groups in the training and test splits of the data set. Structures with lowest and highest RSE are also shown along with their respective RSE values. Average RSE values are calculated over the molecules in the test set. The atoms which form the pattern of the named functional group are highlighted in red

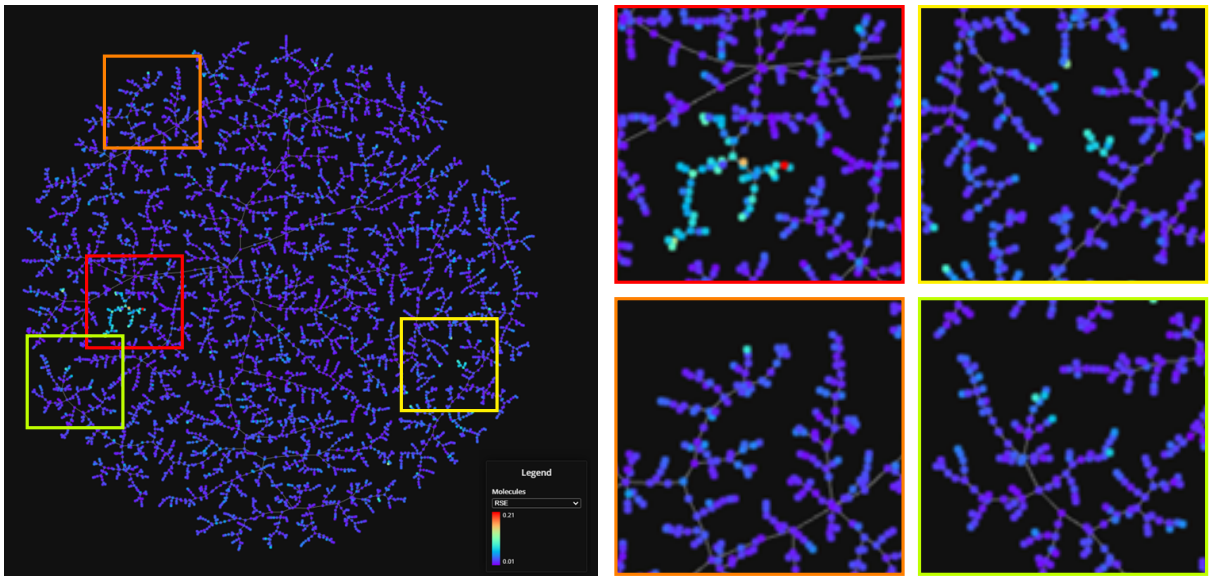
Functional group	**Test %**	**Train %**	**Avg. RSE**	**Min. RSE Structure**	**Max. RSE Structure**
Carboxylic acid	0.91%	0.77%	0.067	 0.0276	 0.2334
Amide	6.44%	6.17%	0.024	 0.0148	 0.2334
Nitrile	12.09%	11.06%	0.024	 0.0137	 0.1666

Continued on the next page...

Functional group	Test %	Train %	Avg. RSE	Min. RSE Structure	Max. RSE Structure
Alkyne	13.45%	11.9%	0.019	 0.0121	 0.1205
Cyclopropane	23.15%	20.84%	0.022	 0.0134	 0.1637

(Dis-)Similarity of hard-to-predict molecules. While the previous analyses focused on the impact of single features of molecules on the performance of GNNs, we also wondered whether hard-to-predict molecules are generally similar to each other or not. To approach this question, we computed pair-wise similarities between all structures in the test set and clustered the resulting matrix using TMAP.[36] Intuitively, the farther two structures appear in the TMAP, the lower their fingerprint similarity, and the more dissimilar their structures. Similarities of a pair of molecules was computed using their Morgan fingerprint.[37] We then colored all molecules according to the RSE obtained when predicting their spectrum, with red nodes having the worst RSE and blue nodes having good RSE, i.e., close to the average (see **Figure. 7**).

A more detailed and interactive version of this TMAP, created with the tool,[39] can be found at,[38] which allows users to zoom in on different regions and structures and to



(a) (b)

Figure 7: a) Full TMAP of RSE values for the SchNet model on test molecules. Each point represents one molecule and the tree structure connects similar molecules. The color coding indicates the RSE values ranging from blue (low) to red (large) according to the color scale. b) Regions where molecules with high RSE values are found. These are often related to one another in chemical structure. An interactive version of this TMAP is available at [38]

inspect the individual structures represented by a node. Clearly, the majority of the tree consists of blue to purple nodes, since most RSE values are concentrated in the lower ranges. Interestingly, however, one can also identify a few clear clusters of red nodes, i.e., sets of highly similar structures that all have rather bad RSE values. The two largest of these hard-to-predict clusters are encircled in red in the figure, and **Figure 9** shows example structures from these encircled regions.

All of the four molecules listed, have a positively charged nitrogen atom, and in the case of the third molecule, is a delocalized carboxylic acid. Since there are few structures with charged entities such as the ones listed in the figure, we hypothesize that the presence of charged features and low frequency count in the data set as compared to other structures, leads to poor performance. The molecular environments in charged entities and delocalized molecules of **Figure 9** differ markedly from their uncharged counterparts, leading to difference in spectra. These high values for structures belonging to a similar family can be

attributed to the lack of enough molecules in the data set, that have such charged atoms.

Problematic molecules in QM9*. As a final note, we would like to mention that a number of molecules in QM9* are unstable structures which deem them unusable for most real life chemical applications and which occur rarely in nature (see, example molecule 4 in the cyclopropane row of **Figure 9**). The reason for this is that QM9* was constructed from GDB-17,[40] a set of structures that were combinatorially generated from their constituent atoms, without considering their overall relevance or stability.

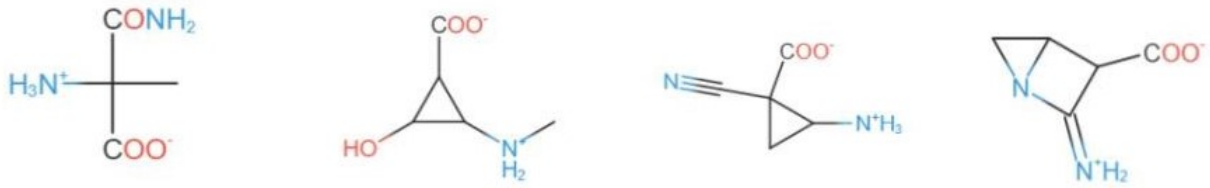


Figure 8: Structures from the highlighted regions showing similarity between the molecules (such as N^+) for which RSE values are much higher than the average RSE.

Conclusion

Spectroscopy has an important place in chemistry and materials research both theoretically and experimentally. Different types of spectra for the same molecule can convey different information about its structure. Therefore, consolidated ML studies on different spectra of the same molecules can fundamentally improve their prediction and bring them a step closer to experimental spectra.

We conduct a benchmark using five leading graph neural networks to predict excitation spectra of organic molecules for the QM9 data set with good accuracy and compare their accuracy to the original spectra computed from density-functional theory. The accuracy of these spectra is determined according to the relative spectral error, which is a parameter specific to spectra. The study finds that the SchNet model performs best on this particular data set, and outperforms previous results by 20% in almost a fraction of the compute

time. More recently developed GNN frameworks could perform even better, as has been demonstrated by their applications to different molecular property datasets.[27,41] There is an increasing demand for creation of molecular property datasets but there are very few approaches to highlight dataset biases or interpret the reasons for good or bad ML predictions on them. Our second contribution is an depth structural analysis of the molecules in the dataset, through structural and functional features, that is crucial to understanding results on any chemistry dataset.

Prediction of spectra using traditional theoretical chemistry methods presents computational challenges, especially in the study of large molecules and nanostructures such as quantum dots[42] and ML progress in this field can definitely help overcome this barrier. Recently, attempts have been made at improving the accuracy of spectra for small molecules which do not require datasets of different molecules, but rather depend on low cost theoretical chemistry calculations of the molecules under investigation.[43,44] However this is still not feasible for large systems. There is a need to incorporate machine learning at a more fundamental level in the formalism of quantum chemical methods, which will alleviate the need of large datasets for prediction of properties. However datasets of spectra will still be crucial if the problem of identifying molecular composition given their spectra has to be tackled using ML, and hence our efforts will be focused on creating datasets that can be used for these tasks.

Conflicts of Interest

The authors declare no conflicts of interest.

Supporting Information

Supporting Information is available online. It contains a detailed mathematical description of the different models that were analysed in this work. Further we enlist the hyperparametrs of the final models for each variant of GNN that was considered in the analysis. In further

sections violin plots regression plots for the GCN,GIN and MPNN are discussed. Finally, we enlist all the different functional groups that were considered for the ease of prediction analysis. This information is available free of charge via the Internet at http://pubs.acs.org.

Acknowledgements

K.S, J.M and L.W. gratefully acknowledge the support received from the HEIBRiDS graduate school. We would like to thank the Helmholtz-Zentrum Dresden-Rossendorf for the provision of computing resources that made this study possible.

Funding information

The research in this manuscript was funded by the HEIBRiDS (Helmholtz Einstein International Research School in Data Science) graduate school. It is jointly funded by the Einstein Center Digital Future (ECDF) and the Helmholtz Association.

References

(1) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. *arXiv:1704.01212 [cs]* **2017**, arXiv: 1704.01212.

(2) Dral, P. O. Quantum Chemistry in the Age of Machine Learning. *The Journal of Physical Chemistry Letters* **2020**, *11*, 2336–2347.

(3) Rupp, M.; von Lilienfeld, O. A.; Burke, K. Guest Editorial: Special Topic on Data-Enabled Theoretical Chemistry. *The Journal of Chemical Physics* **2018**, *148*, 241401.

(4) Ceriotti, M.; Clementi, C.; Anatole von Lilienfeld, O. Introduction: Machine Learning at the Atomic Scale. *Chemical Reviews* **2021**, *121*, 9719–9721.

(5) Brand, J. C. D. *Lines of light: the sources of dispersive spectroscopy*; Gordon Breach Publ.: Luxembourg; United States, 1995.

(6) Thomas, N. C. The early history of spectroscopy. *Journal of Chemical Education* **1991**, *68*, 631.

(7) Marques, M. A. L., Ullrich, C., Nogueira, F., Rubio, A., Burke, K., Gross, E. K. U., Eds. *Time-Dependent Density Functional Theory*; Lecture Notes in Physics; Springer-Verlag: Berlin Heidelberg, 2006.

(8) Zhang, J.; Terayama, K.; Sumita, M.; Yoshizoe, K.; Ito, K.; Kikuchi, J.; Tsuda, K. NMR-TS: de novo molecule identification from NMR spectra. *Science and Technology of Advanced Materials* **2020**, *21*, 552–561.

(9) Ramakrishnan, R.; Hartmann, M.; Tapavicza, E.; von Lilienfeld, O. A. Electronic spectra from TDDFT and machine learning in chemical space. *The Journal of Chemical Physics* **2015**, *143*, 084111.

(10) Ghosh, K.; Stuke, A.; Todorović, M.; Jørgensen, P. B.; Schmidt, M. N.; Vehtari, A.; Rinke, P. Machine Learning: Deep Learning Spectroscopy: Neural Networks for Molecular Excitation Spectra (Adv. Sci. 9/2019). *Advanced Science* **2019**, *6*, 1970053.

(11) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nature Communications* **2017**, *8*, 13890.

(12) Fung, V.; Zhang, J.; Juarez, E.; Sumpter, B. G. Benchmarking graph neural networks for materials chemistry. *npj Computational Materials* **2021**, *7*, 1–8.

(13) David, L.; Thakkar, A.; Mercado, R.; Engkvist, O. Molecular representations in AI-driven drug discovery: a review and practical guide. *Journal of Cheminformatics* **2020**, *12*, 56.

(14) Kovács, P.; Zhu, X.; Carrete, J.; Madsen, G. K. H.; Wang, Z. Machine-learning Prediction of Infrared Spectra of Interstellar Polycyclic Aromatic Hydrocarbons. *The Astrophysical Journal* **2020**, *902*, 100.

(15) Fu, W.; Hopkins, W. S. Applying Machine Learning to Vibrational Spectroscopy. *The Journal of Physical Chemistry A* **2018**, *122*, 167–171.

(16) Zhang, Y.; Ye, S.; Zhang, J.; Hu, C.; Jiang, J.; Jiang, B. Efficient and Accurate Simulations of Vibrational and Electronic Spectra with Symmetry-Preserving Neural Network Models for Tensorial Properties. *The Journal of Physical Chemistry B* **2020**, *124*, 7284–7290.

(17) Westermayr, J.; J. Maurer, R. Physically inspired deep learning of molecular excitations and photoemission spectra. *Chemical Science* **2021**, *12*, 10755–10764.

(18) Yang, Z.; Chakraborty, M.; White, A. D. Predicting Chemical Shifts with Graph Neural Networks. *bioRxiv* **2020**, 2020.08.26.267971.

(19) Kipf, T. N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]* **2017**, arXiv: 1609.02907.

(20) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics* **2018**, *148*, 241722.

(21) Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? *arXiv:1810.00826 [cs, stat]* **2019**, arXiv: 1810.00826.

(22) Weininger, D. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information & Computer Sciences* **1988**, *28*, 31–36.

(23) Montavon, G.; Hansen, K.; Fazli, S.; Rupp, M.; Biegler, F.; Ziehe, A.; Tkatchenko, A.; Lilienfeld, A.; Müller, K.-R. Learning Invariant Representations of Molecules for Atomization Energy Prediction. *Advances in Neural Information Processing Systems* **2012**, *25*.

(24) Hancock, J. T.; Khoshgoftaar, T. M. Survey on categorical data for neural networks. *Journal of Big Data* **2020**, *7*, 28.

(25) Vinyals, O.; Bengio, S.; Kudlur, M. Order Matters: Sequence to sequence for sets. *arXiv:1511.06391 [cs, stat]* **2016**, arXiv: 1511.06391.

(26) Jørgensen, P. B.; Jacobsen, K. W.; Schmidt, M. N. Neural Message Passing with Edge Updates for Predicting Properties of Molecules and Materials. *arXiv:1806.03146 [cs, stat]* **2018**, arXiv: 1806.03146.

(27) Schütt, K. T.; Unke, O. T.; Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv:2102.03150 [physics]* **2021**, arXiv: 2102.03150.

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

(29) RDKit: Open-source cheminformatics. `http://www.rdkit.org`, (Accessed April 16, 2020).

(30) Fey, M.; Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv:1903.02428 [cs, stat]* **2019**, arXiv: 1903.02428.

(31) Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for Pre-training Graph Neural Networks. *arXiv:1905.12265 [cs, stat]* **2020**, arXiv: 1905.12265.

(32) Xu, W. GNN models implemented in pytorch. https://github.com/snap-stanford/pretrain-gnns, (Accessed April 18, 2020).

(33) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* **2017**, arXiv: 1412.6980.

(34) Glavatskikh, M.; Leguy, J.; Hunault, G.; Cauchy, T.; Da Mota, B. Dataset's chemical diversity limits the generalizability of machine learning predictions. *Journal of Cheminformatics* **2019**, *11*, 69.

(35) Daylight Theory: SMARTS - A Language for Describing Molecular Patterns. https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html, (Accessed April 22,2019).

(36) Probst, D.; Reymond, J.-L. Visualization of very large high-dimensional data sets as minimum spanning trees. *Journal of Cheminformatics* **2020**, *12*, 12.

(37) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754.

(38) Interactive TMAP on test data. https://try-tmap.gdb.tools/tmap/soft-garnet-chameleon, (Accessed April 11, 2022).

(39) tmap - Visualize big high-dimensional data. https://tmap.gdb.tools/#mai-qm, (Accessed Februrary 15, 2020).

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

(41) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *Journal of Chemical Theory and Computation* **2019**, *15*, 3678–3693.

(42) Bera, D.; Qian, L.; Tseng, T.-K.; Holloway, P. H. Quantum Dots and Their Multimodal Applications: A Review. *Materials* **2010**, *3*, 2260–2345.

(43) Xue, B.-X.; Barbatti, M.; Dral, P. O. Machine Learning for Absorption Cross Sections. *The Journal of Physical Chemistry A* **2020**, *124*, 7199–7210.

(44) Westermayr, J.; Marquetand, P. Deep learning for UV absorption spectra with SchNarc: First steps toward transferability in chemical compound space. *The Journal of Chemical Physics* **2020**, *153*, 154112.

For Table of Contents only

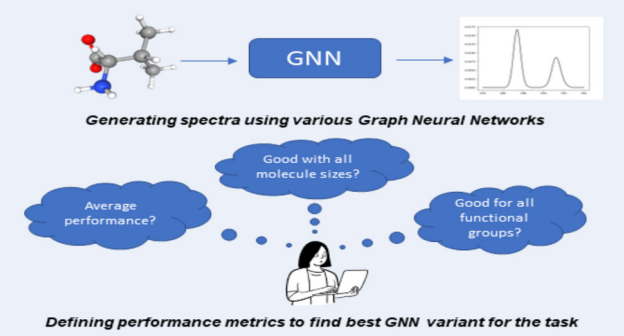


Figure 9: Table of contents entry