Node-aware convolution in Graph Neural Networks for Predicting molecular properties

Linh Le Pham Van
UET AILab, VNU
Hanoi, Vietnam

Quang Bach Tran
UET AILab, VNU
Hanoi, Vietnam

Tien Lam Pham
PIAS, Phenikaa University
Hanoi, Vietnam

Quoc Long Tran
UET AI Lab, VNU
Hanoi, Vietnam

***Abstract*—Molecular property prediction is a challenging task which aims to solve various issues of science namely drug discovery, materials discovery. It focuses on understanding the structure-property relationship between atoms in a molecule. Previous approaches have to face difficulties dealing with the various structure of the molecule as well as heavy computational time. Our model, in particular, utilizes the idea of message passing neural network and Schnet on the molecular graph with enhancement by adding the Node-aware Convolution and Edge Update layer in order to acquire the local information of the graph and to propagate interaction between atoms. Through experiments, our model has been shown the outperformance with previous deep learning methods in predicting quantum mechanical, calculated molecular properties in the QM9 dataset and magnetic interaction of two atoms in molecules approaches.**

***Index Terms*—deep learning, quantum chemistry, graph neural networks**

I. Introduction

Density functional theory (DFT) [1], [2] plays an important role in physics for molecular property prediction. Based on DFT, many techniques have been developed to model the interaction of molecules. However, since DFT simulations are computationally expensive. These methods also hardly employ large molecules with millions of atoms. These drawbacks of DFT promote the development of a new research field, named materials informatics which mainly applies Machine learning methods to present molecular properties. Machine learning, especially deep learning, has triggered a paradigm shift in materials science study, when materials data, including experiment and calculation data, can be accessed easily and freely [3]–[6]. By using the Machine learning approach, one expects to speed the process of discovery of new molecules or materials, in which it requires to utilize the fast estimation of molecular properties and to discover hidden chemistry and physics from data. Despite certain advances, the Machine learning approaches [3]–[6] still carry a weakness which is the overdependence on the pre-processing input data.

To solve the problem of input data representation, many recent studies focus on presenting, developing and improving Graph Neural Networks (GNNs), the Deep Learning models that handle the input data represented as a graph. Therefore, it is possible to apply Graph Neural Networks to solve the task of predicting properties in Quantum Chemistry. [8]–[12] using Graph Neural Networks to solve the task of predicting properties in quantum chemistry have shown significant improvements in speed and accuracy compared to other methods such as DFT or traditional Machine learning [3]–[6].

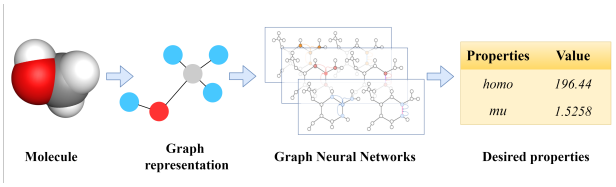

Fig. 1: The pipeline of using Graph Neural Networks for predicting molecular properties. At the first step, molecule is represented as graph format and then feedforwarded through Graph Neural Networks to predict the molecule properties.

In this paper, we focus on solving the drawbacks of some state of the art models, MPNNs [10] and SchNet [11], on improving the accuracy in predicting molecular properties tasks. From that, we propose our model, NAGCN, and demonstrate that it get better accuracy compared with the state of the art model SchNet [11]. We summarize our contribution as follow:

- We generalize the continuous-filter convolution in [11] to Node-aware Convolution and add it to the model, which help to collect more high-level information especially local features in the graph-based dataset.
- We introduce a new Edge Update layer which helps to pass the interaction information in molecules more efficiently.
- We modify the architecture of the Readout layer, allowing our model to use more information from multiple Interaction layer for aggregating output.

The paper is organized as follow: Section 2 describes Related Works, followed by the Proposed Method in Section 3, Section 4 is the Results of experience, and Section 5 is the Conclusion.

II. Related works

To predict molecular properties, Graph Neural To predict molecular properties, Graph Neural Networks learns to model molecular systems from molecular-input data. A common approach to modelling molecular systems is to divide them into local environments when the properties of the molecules are considered to be the sum of all the contributions of each atom. Based on these contributions, the original property is reconstructed through a synthetic layer built on physical

knowledge [7]. As described in the Figure 1, Graph Neural Networks receive input data as a molecular graph and learn the node features vector of each atom in the molecule, then use these feature vectors to calculate the desired properties output, such as the molecular properties (potential energy, force) or the interaction value of atom pairs (J coupling value). The following, we briefly review the related works that will be used in the evaluation of our experiment: Message Passing Neural Networks [10] (MPNNs) and SchNet [11].

Message Passing Neural Networks: Recently, MPNN family models [10] are known as some of the most popular neural networks working efficiently with tasks of predicting molecule properties. All of them have the same formulations, which are: In the first phase, the message and update function take the responsibility to learn features of molecule in high levels feature. After that, the readout function integrates all information in previous steps in order to make the final result for molecule properties. However, these MPNN models [10] have several drawbacks requiring much information in input, leading to time-consuming for carefully choosing feature in the input.

SchNet and Continuous-filter convolutions: The SchNet model [11] was developed and published by Schutt and his team in 2017. The SchNet model learns the hidden representation vectors of atoms showing local contributions by using stacked Interaction layers and sum them up via the Readout layer for calculating the desired output. [11] proposed a continuous-filters convolution and uses this convolution in the Interaction class to update the hidden representation vectors h_i. For a molecule consisting of a set of atomic representation vectors $h_1, h_2, ..., h_N$ has positions $r_1, r_2, ..., r_N$, the continuous-filters convolution updates the atomic representation vector h_i at the update time t by the equation:

$$h_i^t = \sum_{j \in N(i)} h_j^t \circ W(\| r_j - r_i \|) \tag{1}$$

where $\circ$ denote element-wise multiplication, and the $W(\| r_j - r_i \|)$ is the filter generating layer. Using continuous-filters convolution for updating the atomic representation vector h_i, SchNet model can model the local interactions between atoms in the molecule [11]. Through the experiments, SchNet has been shown to achieve better results in predicting molecular properties than the previous MPNNs [10]. However, the SchNet model still has weaknesses that need to be improved. First, the convolution is used to update the atom-specific vector h_i only using information about the distance (spatial information) between the atoms to initiate the weight that allows multiplication. This may not be sufficient to update features vector of atoms. Next, the SchNet model did not mention the characteristic e_{ij} edge vectors between atoms, so it did not update the edge vectors either. Finally, the SchNet model only uses node vectors at the last Interaction layer for the synthesis and prediction process (Readout) the output properties of the molecule. This may cause the model to miss some information from previous layers and make the model not highly accurate.

III. Proposed Method

A. Definition

For simplicity, we describe a molecule as an undirected graph $G = (V, E)$, where V is the set of nodes, and E is the set of edges. In graph G, we denote $h_i \in V$ is the node feature that represents the i-th atom in the molecule, and $e_{ij} \in E$ is the edge feature, representing for the relationship between the i-th and j-th atoms. The node i has the set $N(i)$ containing all neighbours of its.

B. Node-aware convolution

Using the idea of continuous-filters convolution [11], we propose the generalized continuous convolution is Node-aware convolution, and then use it in constructing our model.

Specifically, the hidden vector representation of the atom is updated according to the Node-aware convolution:

$$h_i^{t+1} = \sum_{j \in N(i)} h_j^t \circ f_{ij} \tag{2}$$

where the features vector f_{ij} indicates the relationship between two nodes, i and j. In [11], f_{ij} is a filter generating layer $W(\| r_j - r_i \|)$, which calculates the relationship between two nodes i and j based on the distance between them. Meanwhile, the Node-aware convolution shows that f_{ij} can describe a more general relationship, not just based on the distance relationship between two atoms i, j. Find that the interaction between two atoms in a molecule is based not only on the distance between them but also on the two atoms themselves, we using both distance and relationship between two atoms to calculate the features vector f_{ij} and use this features vector f_{ij} to update atomic representation vector h_i via Eq. 2. We also consider the features vector f_{ij} as the edge vector of the molecular graph and use a special Edge update layer to update them during the model training. From now on, we consider two vectors f_{ij} and e_{ij} as one. Details about the edge vector e_{ij} and the Edge update layer will be presented in the sub-section III-C.

C. Architecture

In this section, we will introduce our model, named NAGCN (Node Aware Graph Convolutional Network), for predicting molecular properties tasks. The architecture of the proposed model is presented in Figure 2, including the main parts that will be discussed below. The input data for the model consists of molecules with a set of nuclear charge z and position r. The input vector initialization process, including Embedding and Spatial generating layer, will initialize the node and edge vectors for the model from the z charge and position r. These vectors are then updated through T stacked representation layers, Interaction layers and Edge update layers. Finally, the output node vectors will be used to aggregate the desired output properties of the molecule via Readout layer.

Constructing the molecular graph

In our model, to build a molecular graph, we use the cutoff function to initialize weights for the edges of the graph model. Using the input as the distance d_{ij} between the two atoms, the cutoff function calculates the weight representing the edge

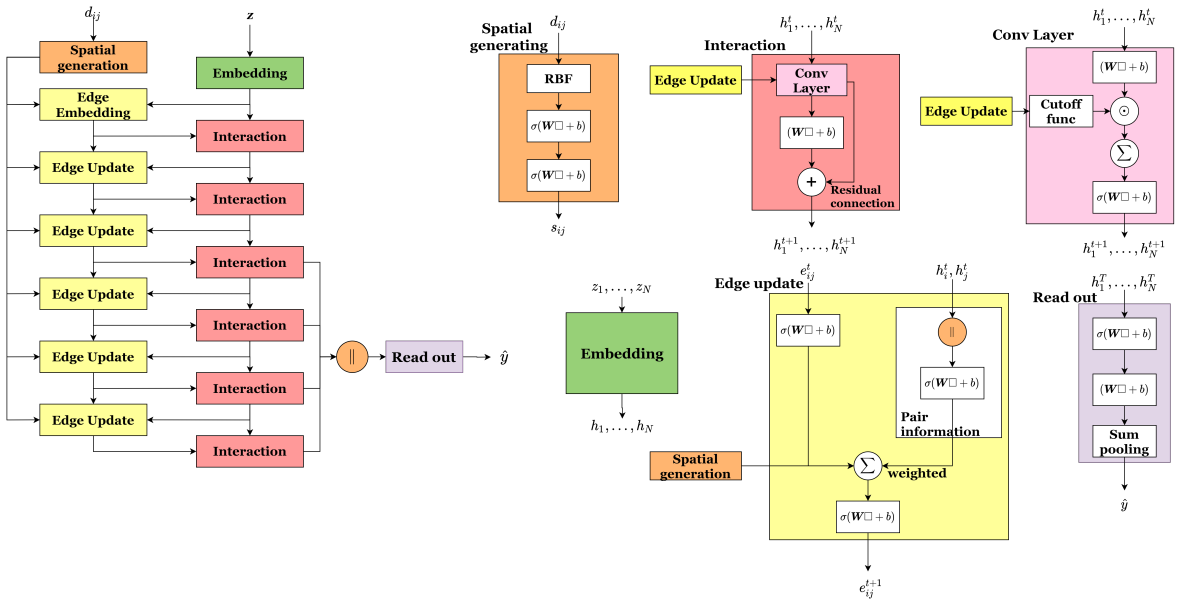

Fig. 2: The architecture of NAGCN. The figure on the left-hand side represents the overview of model and the figure on the right-hand side shows the detailed architecture of each layer in NAGCN.

weight between the two atoms. The edge weight representing the existence of an edge between two atoms is the value in the segment [0, 1], with a value of 0 indicating that there is no edge between the two atoms and the remaining values represent the weight of the edge. This weight is then used to calculate the edge vector during the process of updating vector nodes and edge vectors. Based on suggestion of [14], we use the cosine-cut function presented by the Eq. 3 to help model learn the local interactions in the molecule in the best way.

$$f_{\mathrm{c}}\left(d_{ij}\right)=\begin{cases}\frac{1}{2}\left[1+\cos\left(\frac{\pi d_{ij}}{d_{\mathrm{c}}}\right)\right], & d_{ij}<d_{\mathrm{c}}\\ 0, & d_{ij}\geq d_{\mathrm{c}}\end{cases} \tag{3}$$

Embedding and Spatial generating layers

To model the molecule with as little information as possible, the model uses only the input as a molecular 3D model, with the atoms and their positions in space, to initialize the node and edge vectors.

Specifically, to initialize the atom representation vector in the molecule, we use Embedding layer. An atom with atomic charge z, through Embedding layer, will be initialized to a node vector h_i^0, which is learnable embedding vector and can be updated in training process. The atoms with the same atomic charge will have the same initial representation h_i^0.

The space-specific vector s_{ij} of the molecule is initialized via the Spatial generating layer. The distance d_{ij} between atoms is passed through the RBF function to initialize the vector which carries spatial information in the molecule. This vector is then passed through a fully connected neural network follow by activation function to help the spatial vector become more nonlinear and robust. In our model, we use shifted softplus $\mathrm{ssp}(x)=\ln\left(0.5e^x+0.5\right)$ is activate function because of suggestion in [14].

RBF function is used following suggestion of [11] to expand spatial information between atoms is defined in Eq. (4):

$$RBF\left(d_{ij}\right)=\exp(-\gamma\|d_{ij}-\mu\|) \tag{4}$$

where two hyperparameters γ and μ are selected so that the output vector can carry information about the entire distance between two possible atoms in the dataset.

After the spatial vectors and the node vectors are initialized, the edge vectors are initialized based on the equation:

$$e_{ij}^0=\alpha(W_1 s_{ij})+(1-\alpha)W_2(h_i^0\parallel h_j^0) \tag{5}$$

where s_{ij} is a learnable vector that contains spatial information between the atoms, $W_2(h_i^0\parallel h_j^0)$ denotes the relationship between two atoms i and j and α is a hyperparameter that controls the contribution of the relationship between two atoms to the edge vector. During our experiments, we set α with a value of 0.8. By using Eq. (5), the initial edge vector carries both information about the spatial relationship in the molecule and the relationship between two atoms.

Interaction and Edge update layer

To model the molecule from the structural and spatial information generated from the previous layer, we use stacked Interaction and Edge update layers. These layers are used as the crucial components of our model.

Using the convolution formula presented in the sub-section III-B as a node vector update function, the Interaction layer learns the hidden representation vector of atoms. Specifically, at the t-th Interaction layer, the node feature vector is updated via the Eq. 2 with e_{ij} edge vector that carries information about both space and the relationship between two atoms i and j. In our model, we also use Residual connection [18] liked SchNet [11] for keep the model not overffiting. Besides, edge

vector e_{ij} is also updated via the Edge update function $e_{ij}^{t+1} = E(e_{ij}^t, h_i^t, h_j^t, s_{ij})$. In our work, we use the Edge update class shown in the equation below:

$$e_{ij}^{t+1} = W_1 e_{ij}^t + \alpha W_2 \left(h_i^t \| h_j^t\right) + \beta W_3 s_{ij} \tag{6}$$

where that $W_2 \left(h_i^t \| h_j^t\right)$ is learnable function that learn the relationship between two atoms i and j, $W_3 s_{ij}$ is the vector that contains the spatial information and can be generated by Sptatial generating layer, $W_1 e_{ij}^t$ is the previous edge vector, and α and β are the two hyperparameters control the contribution of information about the relationship between the two atoms and the spatial information of the molecule to the feature vector. By using Edge Update layer, edge vectors in our network can get more information about both spatial relationship in the molecule and the relationship between two atoms and make NAGCN become robust and get better accuracy compared with the state of the art models.

Readout layer

After going through all the Interaction and Edge update classes, we have atom representations at different levels. In order to predict molecular properties, we use Readout layer for aggregating features from all atoms. First of all, the final atom representations are calculated following equation:

$$h_i^* = \sigma W \left(\|_{k=0}^n h_i^k\right) \tag{7}$$

The idea of the Eq. 7 is that we not only use the atom representations from last Interaction layer, but also use the atom representations from previous layers for predict desired properties. The effect of using many Interaction layers for calculating output will be shown in sub-section IV-A1. After that, node features vector of all atoms is sum up by using sum pooling functions like the suggestion of [14] to calculate output property. The sum pooling function is invariant to premutations of the node so it makes our model to be invariant to graph isomorphism.

IV. Experiments

We conduct experiments on two predictive tasks. The first is the task of predicting the J coupling constant between atomic pairs in the molecule, organized by Kaggle [15]. After proving the capabilities of our model on the new task, we conducted experiments on the standard benchmark dataset in quantum chemistry, QM9 [16], [17], to prove that NAGCN is also more accurate than base model on predicting molecular properties tasks.

A. Dataset

1) J coupling dataset: The J coupling dataset, provided by Kaggle [15] with the aim to create the dataset for training the models which can calculate the magnetic interaction between every atom-pairs in molecular. The dataset includes data on 7,164,264 J coupling pairs with eight types of 130,789 molecules along with their molecular structures. The J coupling dataset is divided into two separate train and test sets, with the corresponding dimensions of 4,659,075 J coupling pairs of 85,012 molecules and 2,505,189 J coupling pairs of 45,777 molecules. Along with that, information about these additional attributes is also provided in this dataset. Because the values of J coupling pairs are only published in the train set, we used the train dataset to conduct our experiments.

2) QM9: QM9 [16], [17] is a standard dataset, widely used to evaluate various models for predicting molecular properties tasks. The QM9 dataset consists of more than 130k organic molecules with 13 properties, made up of up to 9 heavy atoms, C, O, F, N, belong to the GDB 17 chemical universe including more than 166 billion parts organic.

B. Experiment setup

In order to conduct training and evaluation of the model, we split the data set into three smaller datasets train, test and vaild, with the proportions of 8:1:1 for J coupling dataset and 110k:10k:10k for the QM9 dataset.

We choose the MSE function as the loss function for the training, MAE, LogMAE for evaluation. To train the models, we used a mini-batch stochastic gradient descent with ADAM optimizer. Batch size is selected as 100 and learning rate is initialized in the range of 1e-3 to 1e-5. We conduct training specific models for each type of molecular properties.

C. Results

1) Predicting J coupling constant: In this subsection, we will show the improvements of our model by comparing its accuracy with the SchNet.

Model modifications for J coupling task

[13] indicates that each J coupling constant of an atomic pair can be divided into the component contributions of each atom in the molecule. Therefore, it is possible to use graph neural network models to learn the features vector of each atom, then use these vectors to synthesize the desired output value - J coupling constant value.

To predict the J coupling pair, [13] uses the Pseudo Labeling method to mark the two atoms containing the J coupling pair. Different from the method of [13], we mark the atomic pair with J coupling to predict with index 2 and the remaining other atoms in the molecule with the index 1. Then, the indices of each atom is passed through an Embedding layer to initialize the vector that carries the information about the J coupling pair to predict. This embedding vector is then concatenated to the embedding vector initialized by nuclear charge z to get the node initialize vector h_i^0.

In addition to separating the atomic charge and the number of atoms (which belong to the J coupling pair or not) for the process of initialization of the features vector of atoms, we also use additional the two auxiliary branches for predicting the properties related with the J coupling value are mulliken charge and four J coupling contributions (fc, sd, pso, dso). The use of two auxiliary branches serves as a regularization method for the model. Due to the use of auxiliary branches, the loss function used in the J coupling prediction task is a combined loss function, given by Eq. (8):

$$L = MSE_{Jcoupling} + \alpha MSE_{mul} + \beta MSE_{4contrib} \tag{8}$$

TABLE I: Predictive performance of the two models in J coupling constant task prediction.

	MAE			*LogMAE*		
	Best single model	**Ensemble model**	**NAGCN +mul+4contrib**	**Best single model**	**Ensemble model**	**NAGCN +mul+4contrib**
1JHN	0.1315	0.1212	**0.1161**	-2.0286	-2.1101	**-2.1529**
1JHC	0.1896	**0.1711**	0.1879	-1.6620	**-1.7657**	-1.6721
2JHN	0.0526	**0.0463**	0.0525	-2.9420	**-3.0734**	-2.9473
2JHC	0.0817	0.0755	**0.0675**	-2.5043	-2.5842	**-2.6961**
2JHH	0.0404	0.0368	**0.0365**	-3.2102	-3.3013	**-3.3107**
3JHN	0.0486	0.0420	**0.0417**	-3.0230	-3.1702	**-3.1772**
3JHC	0.0940	0.0887	**0.0841**	-2.3643	-2.4227	**-2.4755**
3JHH	0.0406	**0.0351**	0.03699	-3.2033	**-3.3486**	-3.2972

TABLE II: Evaluation results of 3 models NAGCN1, NAGCN4 and NAGCN7.

	NAGCN1	**NAGCN4**	**NAGCN7**
MAE	0.1336	**0.1304**	0.1969
LogMAE	-2.0125	**-2.0368**	-1.6250

where α and β are two hyperparameters that control the trade-off between model accuracy in predicting J coupling values and predicting sub-properties values. In our works, we set α is 2 and β is 1. Experiments bellow will show the efficient of auxiliary branches to help the model achieve better accuracy.

Number of interaction and embedding layers used for output aggregation

We conducted an experiment to test the idea of using the additional output from the multiple Interaction classes shown in the section. We compared the accuracy of the three NAGCN models with the number of different Interaction layers used to aggregate the output. The models in turn use 1 Interaction layer (NAGCN1), 4 Interaction layers (NAGCN4) and all Interaction and Embedding classes (NAGCN7) to synthesize the output. As shown in Table II, NAGCN4 model has the highest accuracy compared to the other two models. This suggests that the use of additional vertices vectors from the interaction layers near the end helps the model have more information for the aggregate output. However, when using both vector nodes in the first layers, the accuracy of the model decreases. This is explained by the fact that the node vectors in the first layers are not high-level features. Therefore, adding these vectors is similar to adding noise to the model and reducing accuracy.

Effects of auxiliary branch

To evaluate the effect of the auxilary branches on the model's results, we compared the performance of SchNet [13], NAGCN4, NAGCN4 models with mulliken charge (NAGCN4+mul) and NAGCN4 with Mulliken charge and scalar distribution (NAGCN4+mul+4contrib). We selected the data set of type J coupling 1JHN for experiment. Table III shows that the NAGCN4 model achieves better performance than the SchNet model. Besides, when using auxiliary branches, the accuracy continues improving. This shows that the auxiliary branches helps the model improves accuracy.

Predictive performance in all dataset

Because of the improvements of using multiple Interaction classes for predicting output and using auxilary branches for Regularization, we compared NAGCN+mul+4contrib model with SchNet model [13] experimented on predicting J coupling constant task. [13] conduct training on many models and conduct ensemble to be modeled with higher accuracy. Due to hardware constraints, we do not conduct ensemble of re-proposed models. Table I shows the results of the proposed model with the SchNet best model and the Ensemble model of [13]. Compared with the best single model, the NAGCN model surpasses all J coupling types. Besides, when compared to the ensemble model, NAGCN also achieved better results in 5 out of 8 J coupling types. This result shows the potential and ability to improve the accuracy of NAGCN model compared to SchNet model in J coupling prediction task. We believe that, if there is enough hardware needed, the Ensemble NAGCN model will outperform the Ensemble SchNet model.

TABLE III: Evaluation results of 4 models, SchNet baseline model and three NAGCN models using various number of auxiliary branches.

	SchNet	**NAGCN**	**NAGCN +mul**	**NAGCN +mul+4contrib**
MAE	0.1510	0.1304	0.1279	**0.1161**
LogMAE	-1.889	-2.0368	-2.0565	**-2.1529**

TABLE IV: Predictive accuracy of NAGCN and baseline models on the QM9 dataset.

Properties	**Unit**	**enn-s2s**	**SchNet**	**SchNet EdgeUpdate**	**NAGCN**
Cv	Kcal/mol	0.0400	0.0310	0.0320	**0.0307**
zpve	meV	1.5	**1.47**	1.49	1.49
gap	eV	0.069	0.0711	0.058	**0.0543**
U0	eV	0.019	0.0105	0.0105	**0.0091**
H	eV	0.017	0.0104	0.0113	**0.0090**
homo	eV	0.043	0.0442	0.0367	**0.0342**
r2	Bohr**2	0.18	0.0713	0.072	**0.0590**
U	eV	0.019	0.0106	0.0106	**0.0092**
G	eV	0.019	0.011	0.0122	**0.010**
alpha	Bohr**3	0.092	0.075	0.077	**0.0725**
lumo	eV	0.037	0.0354	0.0308	**0.0268**
mu	Debye	0.033	0.044	0.029	**0.0169**

2) QM9: The first experiment has shown that the model is effective in predicting J coupling constant. In this section, we evaluate the model on the QM9 dataset, that is the standard

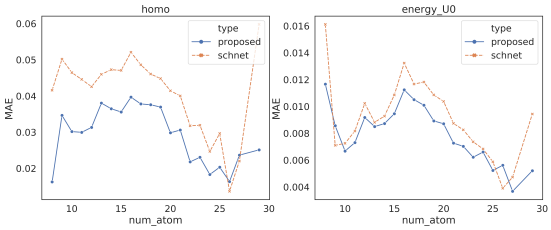

Fig. 3: MAE loss of models in different molecular groups. On the left-hand side is the loss of NAGCN and SchNet on homo property and the right one is the loss of these models on internal energy at 0K.

dataset for the problem of predicting molecular properties.

Predictive performance

We compared the proposed model NAGCN to the base models, including enn-s2s [10], SchNet [11], SchNet with Edge update [12]. As illustrated in Table IV, NAGCN is more accurate at 11 of the 12 properties predicted on the QM9 set than the baseline models. Specifically, compared with the SchNet [11], NAGCN improves the MAE error from 3.3% to 23.6%. This result shows that NAGCN improves accuracy compared to the SchNet. In addition, compared to the SchNet with Edge update [12], the NAGCN model also exhibits superiority with lower MAE errors across all 12 properties. This also demonstrates the use of spatial information at each Edge update layer makes the model work more efficiently than using only spatial information at the first layer like Jorgensen's model [12].

We also experimented with fault analysis of the model as the number of atoms in the molecule increased. Figure 3 shows the error comparison results of models on each molecular group. We chose the state of the art model, SchNet and two molecular properties (U0 and homo) to conduct experiments. The results show that the NAGCN model has lower errors than the SchNet model on most molecular groups. Along with that, when the number of atoms increases, SchNet tends to increase errors fast but NAGCN does not face with this problem.

Generalizability

The number of molecules in chemistry is huge, but the amount of labeled data is limited. Therefore, generalizability is an important factor when evaluate models. We compared the accuracy of NAGCN and SchNet, when trained on data sets with sizes of 50k, 100k and 110k molecules. Table V shows that the NAGCN model achieves better accuracy than the SchNet model even when trained with small datasets. This shows a good generalization ability of NAGCN model compared to SchNet.

TABLE V: The comparison performance on QM9 dataset with different sizes.

	50,000	*100,000*	*110,000*
SchNet	0.0668	0.0485	0.0442
NAGCN	**0.0544**	**0.0342**	**0.0342**

V. CONCLUSION

We have proposed the NAGCN - an deep architecture for predicting molecular properties in Quantum Chemistry. Our model is the extension of SchNet architecture with better performance by integrating the Node-aware convolution, new Edge update and some modification of Readout layer. Experiment results on the both J coupling and QM9 dataset shows significant improvement in comparison with other baselines, SchNet [11] and MPNNs [10]. In the future, we wish to extend NAGCN to other dataset, and apply it into another field, such as some point cloud problems in Computer vision.

ACKNOWLEDGMENT

This work has been supported by VNU University of Engineering and Technology.

REFERENCES

[1] Hohenberg, P. and Kohn, W. "Inhomogeneous Electron Gas", American Physical Society, https://link.aps.org/doi/10.1103/PhysRev.136.B864
[2] Kohn, W. and Sham, L. J. "Self-Consistent Equations Including Exchange and Correlation Effects", American Physical Society.
[3] Rupp, Ramakrishnan, Lilienfeld, "Machine learning for quantum mechanical properties of atoms in molecules", The Journal of Physical Chemistry Letters, 6(16):3309–3313, 2015.
[4] Rupp, Tkatchenko, Mu¨ller, Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning", arXiv:1109.2618.
[5] Faber, Hutchison, Huang, Gilmer, Schoenholz, Dahl, Vinyals, Kearnes, Riley, Lilienfeld, "Fast machine learning models of electronic and energetic properties consistently reach approximation errors better than DFT accuracy", arXiv preprint arXiv:1702.05532, 2017.
[6] Hansen, Biegler, Ramakrishnan, Pronobis, Lilienfeld, Muller, Tkatchenko, "Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space", The journal of physical chemistry letters.
[7] J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials", J. Chem. Phys., 134(7):074106, 2011.
[8] Duvenaud, Maclaurin, Iparraguirre, Bombarell, Hirzel, Aspuru-Guzik, Adams, "Convolutional networks on graphs for learning molecular fingerprints", NIPS, pages 2224–2232, 2015.
[9] Kearnes, McCloskey, Berndl, Pande, Riley, "Molecular graph convolutions: moving beyond fingerprints", Journal of Computer-Aided Molecular Design, 30(8):595–608, 2016.
[10] Gilmer, S.Schoenholz, F.Riley, Vinyals, E.Dahl, "Neural Message Passing for Quantum Chemistry", 2017.
[11] Schütt, Kindermans, Sauceda, Chmiela, Tkatchenko, Müller, "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions", 1706.08566.
[12] Jørgensen, Jacobsen, Schmidt, "Neural message passing with edge updates for predicting properties of molecules and materials".
[13] Tony Y., "22nd place solution - Vanilla SchNet", "Kaggle", September. 2019. [Online]. Available: https://www.kaggle.com/c/champs-scalar-coupling/discussion/106424
[14] Schütt, Tkatchenko, Müller, "Learning representations of molecules and materials with atomistic neural networks", 2018.
[15] (2019, May): "CHAMPS Scalar Coupling", Retrieved from kaggle.com/c/champs-scalar-coupling/overview.
[16] Ruddigkeit, Deursen, Blum, Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17", J. Chem. Inf. Model. 52, 2864–2875, 2012.
[17] Ramakrishnan, Dral, Rupp, Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules", Scientific Data 1, 140022, 2014.
[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", Computer Vision and Pattern Recognition 2016.