Directed Message Passing Based on Attention for Prediction of Molecular Properties

Chen Gong, Yvon Maday

Directed Message Passing Based on Attention for Prediction of Molecular Properties

Gong CHEN *[1] and Yvon MADAY †[1,2]

[1]Sorbonne Université, CNRS, Université Paris Cité, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France
[2]Institut Universitaire de France

May 11, 2023

Abstract

Molecular representation learning (MRL) has long been crucial in the fields of drug discovery and materials science, and it has made significant progress due to the development of natural language processing (NLP) and graph neural networks (GNNs). NLP treats the molecules as one dimensional sequential tokens while GNNs treat them as two dimensional topology graphs. Based on different message passing algorithms, GNNs have various performance on detecting chemical environments and predicting molecular properties. Herein, we propose Directed Graph Attention Networks (D-GATs): the expressive GNNs with directed bonds. The key to the success of our strategy is to treat the molecular graph as directed graph and update the bond states and atom states by scaled dot-product attention mechanism. This allows the model to better capture the sub-structure of molecular graph, i.e., functional groups. Compared to other GNNs or Message Passing Neural Networks (MPNNs), D-GATs outperform the state-of-the-art on 13 out of 15 important molecular property prediction benchmarks.

1 Introduction

Applications of molecular mechanics (MM) in the field of drug discovery [1, 2] and materials science [3, 4] allows for the selection of the potential molecules in the vast chemical space to reduce the experimental cost. However, the complexity of the numerical simulations required for accurate enough approximations of the solution to such molecular dynamics leads

*gong.chen@sorbonne-universite.fr
†yvon.maday@sorbonne-universite.fr

to still long numerical processes. Therefore, in the past decades, prediction of the molecular properties using empirical methods or machine learning has been popular.

Compared with traditional molecular fingerprint-based models [5, 6], nowadays we have numerous molecular representation learnings (MRL) that allow for better performances [7–9]. Thanks to the advances in natural language processing (NLP) [10–12], molecules can be treated as one dimensional sequential strings [13–15], such as SMILES [16]. On the one hand, this requires the understanding of the grammar in such a chemical language (e.g. the parentheses in SMILES "CC(C)C" indicate the presence of a second chain). On the other hand, it is inconvenient to corporate SMILES with two dimensional structure information.

In 1997, Sperduti et al.[17] first applied neural networks (NNs) to directed acyclic graphs, which motivated early studies on graph neural networks (GNNs). Later, Gori et al.[18] and Scarselli et al.[19] proposed the outline of GNNs. As these early studies are based on recurrent neural networks, they suffered from expensive computational costs.

Inspired by the success of convolutional neural networks in computer vision [20, 21], graph convolutional networks have been proposed by Kipf et al.[22]. Encouraged by the application of attention mechanism in NLP, graph attention neworks (GATs) has been proposed in 2007 [23]. Since molecules can naturally be represented by molecular graphs, with atoms as nodes and bonds as edges, GNNs have also shown promising results in many related tasks, such as molecular property predictions [7, 8, 24] and molecule graphs generation [25, 26].

General GNNs follow the framework of Message Passing Neural Networks (MPNNs) [7] and model's performance is determined by the specific message passing algorithm. Inspired by Directed MPNN (D-MPNN) [27], we apply the bond-based message passing algorithm but with the scaled dot-product attention mechanism for updating bond states and atom states. To do the molecule-level tasks, there is one attention-based readout function at the end of each layer. Our model, called Directed Graph Attention Networks (D-GATs), consists of 3 parts:

1) Backbone. A chemical bond between two atoms is considered as two different directed bonds in the networks. Based on the scaled dot-product attention mechanism, the directed bonds aggregate neighbors' information and are then used to update the atoms representations. The graph-level representation is a virtual atom embedding [28], updated by a readout function.

2) Pre-training. To alleviate possible overfitting due to small benchmark datasets containing only thousands of molecules, we collected all molecules that appeared in the experimental parts plus the ZINC-250K dataset [29] to compose the pre-training dataset. For the pre-training tasks, in addition to the masked atom prediction task, we also included molecular properties prediction task for molecules from the ZINC-250K dataset, to train the virtual node.

3) Fine-tuning. For a specific downstream task, we just need the feed-forward neural networks to evaluate the graph-level molecular properties.

The pre-training and fine-tuning scheme alleviates the overfitting from limited supervised data and increases the models' performance [24].

In particular, this paper presents the following contributions:

- D-GATs follow the common framework of MPNNs and explore a bond-level message passing algorithm completely relying on scaled dot-product attention mechanism, which outperforms state-of-the-art (SOTA) baselines on 13/15 molecular property prediction tasks (see Table 3 and 4)on the MoleculeNet benchmark [30].
- Propose a simple but efficient pre-training strategy (see section 3.2).
- The code and pre-trained models of D-GATs are publicly available at `https://github.com/GongCHEN-1995/D-GATs`.

2 Neural Network Architecture

In this section, we present the D-GATs framework by explaining the details of the message passing algorithm and readout function. Additionally, we compare our framework with other similar works.

Functional groups (like alcohols, ethers, aldehydes, ketones, carboxylic acids, etc.) play a crucial role in conferring certain physical and chemical properties to the molecule that has them. Hence, developing an efficient algorithm to distinguish such sub-structures is essential for improving the performance of MRL.

Traditional models treat molecular graphs as undirected graphs. In [27], D-MPNN proposed directed bonds to avoid unnecessary loops during the message passing phase of the algorithm. According to the research in [31], one key factor leading to the over-smoothing issue is the over-mixing of information and noise because the interaction message from other atoms may be either helpful information or harmful. From this point, the directed bonds alleviate the over-mixing of information.

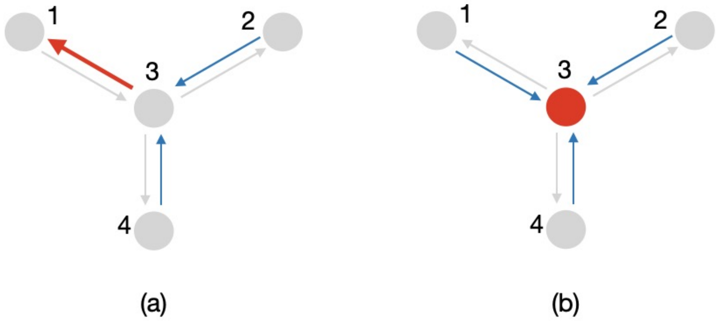
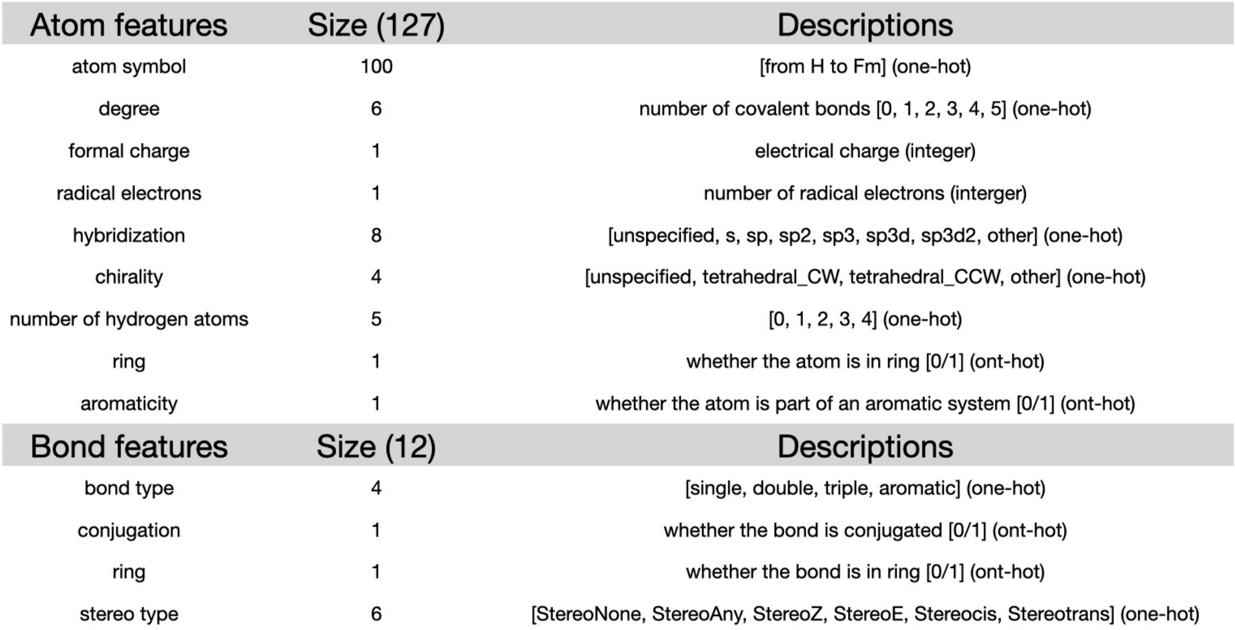


Figure 1: **(a)** Update of bond states. The edge 3 → 1 is updated by (edge 2 → 3 and edge 4 → 3) **(b)** Update of atom states. The node 3 is updated by edge 1 → 3, edge 2 → 3 and edge 4 → 3

But D-MPNN has been only applied the simple aggregate functions, which limits models' performance. In order to improve the performances, we use scaled dot-product attention [10] to aggregate messages. There are two similar works, both using attention mechanism on directed molecular graphs and here are the differences between their models and ours: In [32], GEA is based on additive attention mechanism, which is less efficient than dot-product attention. GEA also tested max-pooling, sum-pooling and set2set [33] as the readout function while our model D-GATs use supervirtual node, a more robust structure. Another model DGANN presented in [34] has similar update function but with totally different logics. DGANN first updates the directed bonds and only the outputs at last layer would be used to update atom states and molecule-level representations. In our model, the bond states, atom states and supervirtual node representations are updated in each interaction layer thus they are tightly coupled.

2.1 Initialization of Input Features

Table 1 lists the required input features to D-GATs. Since categorical data contains label values that cannot be directly processed by our model, we employ one-hot encoding to convert categorical data to numerical data.



Atom features	Size (127)	Descriptions
atom symbol	100	[from H to Fm] (one-hot)
degree	6	number of covalent bonds [0, 1, 2, 3, 4, 5] (one-hot)
formal charge	1	electrical charge (integer)
radical electrons	1	number of radical electrons (interger)
hybridization	8	[unspecified, s, sp, sp2, sp3, sp3d, sp3d2, other] (one-hot)
chirality	4	[unspecified, tetrahedral_CW, tetrahedral_CCW, other] (one-hot)
number of hydrogen atoms	5	[0, 1, 2, 3, 4] (one-hot)
ring	1	whether the atom is in ring [0/1] (ont-hot)
aromaticity	1	whether the atom is part of an aromatic system [0/1] (ont-hot)
Bond features	**Size (12)**	**Descriptions**
bond type	4	[single, double, triple, aromatic] (one-hot)
conjugation	1	whether the bond is conjugated [0/1] (ont-hot)
ring	1	whether the bond is in ring [0/1] (ont-hot)
stereo type	6	[StereoNone, StereoAny, StereoZ, StereoE, Stereocis, Stereotrans] (one-hot)

Table 1: Inputed Atom and Bond Features

Given the input atom features $F^n = \{F_1^n, F_2^n, ..., F_N^n\}, F_i^n \in \mathbb{R}^{127}, i = 1, ..., N$ and the input bond features $F^e = \{F_1^e, F_2^e, ..., F_E^e\}, F_p^e \in \mathbb{R}^{12}, p = 1, ..., E$, where

- n and e in superscript represent atoms and bonds.
- N is the number of atoms in molecule and 127 is the number of possible atom features.

- E is the number of bonds and 12 is the number of possible bond features. We write $p = p(i, j)$ to indicate the bond p that links atoms i and atom j. Note that $p(i, j) = p(j, i)$.

Initialization of directed bonds states: we construct the initial directed bond states from atom i to atom j as:

$$h^0_{\vec{p}(ij)} = W^e_T([F^n_i, F^e_{p(i,j)}, F^n_j]) \tag{1}$$

where [...] is the concatenation operation and $W^e_T \in \mathbb{R}^{D_h \times 266}$ is a learnable matrix to convert the concatenation of $F^n_i, F^e_{p(i,j)}$ and F^n_j into a vector in dimension D_h. D_h is the dimension of model and in our model $D_h = 512$. Even though $F^e_{p(i,j)}$ does not contain any directionality, the two inputed atom features F^n_i and F^n_j cannot be commuted and thus traduce directionality by indicating the start atom and the end atom correspondingly. Thus $h^0_{\vec{p}(ij)} \neq h^0_{\vec{p}(ji)}$.

Initialization of atom states: the initial atom states $h^0 = \{h^0_i | i = 1, 2, ..., N\}$ are transformed from atom features F^n:

$$h^0_i = W^n_T F^n_i \tag{2}$$

where $W^n_T \in \mathbb{R}^{D_h \times 127}$ is a learnable matrix to convert the atom features into a vector in dimension D_h

Initialization of Molecular representations: we introduce a molecular feature, following the notion of supervirtual node $\mathcal{S}$ introduced in Attentive FP [35] that connects all atoms of the molecule. The initialized molecular representation $\mathcal{S}^0 \in \mathbb{R}^{D_h}$ is a trainable vector used to represent molecule and will be updated with attention mechanism.

2.2 Update of Representations

In this subsection, we will talk about how to update the states through scaled dot-product attention mechanism. The update follows the order showed in Figure 2(b). In each interaction layer, we apply three times attention mechanism to update directed bond states, atom states and molecular representations separately. The trainable parameters in layer $t+1$ for attention mechanism are:

$$W^{t+1}_{Q^e}, W^{t+1}_{K^e}, W^{t+1}_{V^e}, W^{t+1}_{Q^n}, W^{t+1}_{K^n}, W^{t+1}_{V^n}, W^{t+1}_{Q^{\mathcal{S}}}, W^{t+1}_{K^{\mathcal{S}}}, W^{t+1}_{V^{\mathcal{S}}} \in \mathbb{R}^{D_h \times D_h}$$

The trainable parameters in multilayer perception (MLP) are:

$$W^e_1, W^e_2, W^n_2, W^n_2, W^{\mathcal{S}}_1, W^{\mathcal{S}}_2 \in \mathbb{R}^{D_h \times D_h}$$

$\sigma(\cdot)$ is the Rectified Linear Unit (ReLU) activation function.

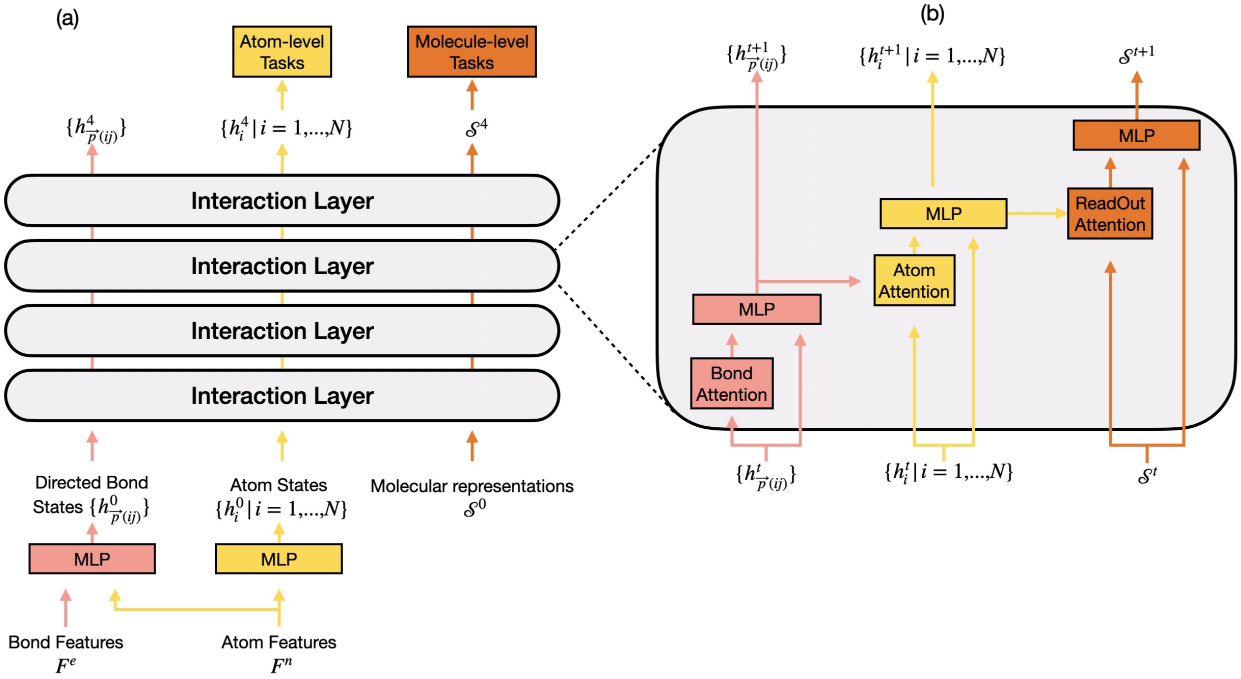


Figure 2: **(a)** Framework of D-GATs with 4 layers. **(b)** Details in each interaction layer

2.2.1 Update of Directed Bond States

Note $\mathcal{E} = \{\vec{p}(ij)\} \cup \{\vec{p}(ki) | k \in \mathcal{N}(i), k \neq j\}$. Following the framework and notations in [7, 27], we compute the bond messages $m^{t+1}_{\vec{p}(ij)}$ by the equations:

$$m^{t+1}_{\vec{p}(ij)} = M^{t+1}_e(h^t_q | q \in \mathcal{E}) = \sum_{q \in \mathcal{E}} \alpha^{t+1}_{\vec{p}(ij),q}(h^t_q W^{t+1}_{V^e}) \tag{3}$$

where $\mathcal{N}(i)$ denotes the neighbor atoms of atom i. The attention-based message functions M^{t+1}_e compute the coefficients $\alpha_{\vec{p}(ij),q}$ $(q \in \mathcal{E})$ by:

$$\alpha^{t+1}_{\vec{p}(ij),q} = \texttt{Softmax}(e^{t+1}_{\vec{p}(ij),z} | z \in \mathcal{E}) = \frac{\exp(e^{t+1}_{\vec{p}(ij),q})}{\sum_{z \in \mathcal{E}} \exp(e^{t+1}_{\vec{p}(ij),z})} \tag{4}$$

$$e^{t+1}_{\vec{p}(ij),q} = \frac{(h^t_{\vec{p}(ij)} W^{t+1}_{Q^e})(h^t_q W^{t+1}_{K^e})^T}{\sqrt{D_h}} \tag{5}$$

Next is a MLP where the messages are used to update directed bond states by update functions U^{t+1}_e:

$$h^{t+1}_{\vec{p}(ij)} = U^{t+1}_e(h^t_{\vec{p}(ij)}, m^{t+1}_{\vec{p}(ij)}) = W^e_2(\sigma(W^e_1(\texttt{LayerNorm}(h^t_{\vec{p}(ij)} + m^{t+1}_{\vec{p}(ij)})))) \tag{6}$$

And `LayerNorm` is from [36].

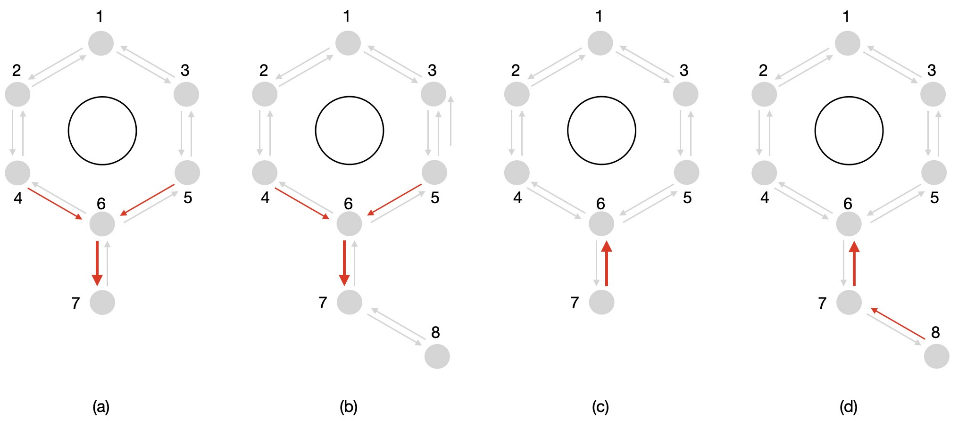


Figure 3: **Example of directed message flow.** **(a)** and **(b)**: $h^{t+1}_{\vec{p}(67)}$ is updated by $[h^t_{\vec{p}(67)}, h^t_{\vec{p}(46)}, h^t_{\vec{p}(56)}]$, thus they have the same embeddings for $t \leq 7$. **(c)** and **(d)**: $h^t_{\vec{p}(76)}$ are different for $t > 0$ due to the existence of $h^t_{\vec{p}(87)}$

Compared to undirected graphs, directed graphs prevent the information from being repeatedly passed back to its source and thus reduce noise. Besides, unless the atom information going through a ring structure, the same substructures always result in the same bond states. For instance, in Figure 3 (a) and (b), h^t_{67} in two molecules are the same if $t \leq 7$. When $t \geq 8$, i.e., with 8 layers, $h^t_{\vec{p}(67)}$ are different in (a) and (b) because the influence of atom 8 arrives at $h^t_{\vec{p}(67)}$ through the chain $8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 6$ after t=8 steps.

Additionally, the computational cost for directed graphs is quadrupled because the number of bonds is doubled (one undirected bond generates two directed bonds).

2.2.2 Update of Atom States

Followed by the update of directed bond states, atom messages m_i^{t+1} are updated through vertex message functions M_n^{t+1}:

$$m_i^{t+1} = M_n^{t+1}(h_i^t, h^{t+1}_{\vec{p}(ji)}|j \in \mathcal{N}(i)) = \alpha^{t+1}_{i,i}(h_i^t W^{t+1}_{V^n}) + \sum_{j \in \mathcal{N}(i)} \alpha^{t+1}_{i,j}(h^{t+1}_{\vec{p}(ji)} W^{t+1}_{V^n}) \tag{7}$$

For $j \in \mathcal{N}(i) \cup \{i\}$, the attention weights are computed as:

$$\alpha^{t+1}_{i,j} = \texttt{Softmax}(e^{t+1}_{i,k}|k \in \mathcal{N}(i) \cup \{i\}) = \frac{\exp(e^{t+1}_{i,j})}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(e^{t+1}_{i,k})} \tag{8}$$

$$e^{t+1}_{i,k} = \begin{cases} \frac{(h_i^t W^{t+1}_{Q^n})(h_i^t W^{t+1}_{K^n})^T}{\sqrt{D_h}} & \text{k=i} \\ \frac{(h_i^t W^{t+1}_{Q^n})(h^{t+1}_{\vec{p}(ki)} W^{t+1}_{K^n})^T}{\sqrt{D_h}} & \text{k} \neq \text{i} \end{cases} \tag{9}$$

Next is to update atom states in MLP:

$$h_i^{t+1} = U_n^{t+1}(h_i^t, m_i^{t+1}) = W_2^n(\sigma(W_1^n(\texttt{LayerNorm}(h_i^t + m_i^{t+1})))) \tag{10}$$

As presented in Figure 1(b), during the update process, atom states collect the information flows in and are independent to the information flows out. The atom states are used to update molecular representation $\mathcal{S}^{t+1}$.

Moreover, as the atom states merge atoms' chemical environment, they can also be applied to do atom-level tasks (e.g. to classify atom type) or to recover masked atoms in pre-training stage.

2.2.3 Update of Molecule Representations

Known the updated atom states h_i^{t+1}, the molecular representations $\mathcal{S}^{t+1}$ are updated by the messages defined as:

$$m^{t+1} = \texttt{ReadOut}^{t+1}(\mathcal{S}^t, h_i^{t+1} | i = 1, 2, ..., N) = \alpha_{\mathcal{S}}^{t+1}(\mathcal{S}^t W_{V\mathcal{S}}^{t+1}) + \sum_{j=1}^{N} \alpha_i^{t+1}(h_i^{t+1} W_{V\mathcal{S}}^{t+1}) \tag{11}$$

for $i \in [1, N] \cup \{\mathcal{S}\}$:

$$\alpha_i^{t+1} = \texttt{Softmax}(e_k^{t+1} | k \in [1, N] \cup \{\mathcal{S}\}) = \frac{\exp(e_i^{t+1})}{\sum_{k \in [1,N] \cup \{\mathcal{S}\}} \exp(e_k^{t+1})} \tag{12}$$

$$e_i^{t+1} = \begin{cases} \frac{(\mathcal{S}^t W_{Q\mathcal{S}}^{t+1})(\mathcal{S}^t W_{K\mathcal{S}}^{t+1})^T}{\sqrt{D_h}} & \text{k}=\mathcal{S} \\ \frac{(\mathcal{S}^t W_{Q\mathcal{S}}^{t+1})(h_k^{t+1} W_{K\mathcal{S}}^{t+1})^T}{\sqrt{D_h}} & \text{k} \in \text{[1,N]} \end{cases} \tag{13}$$

Finally, the molecular representations are:

$$\mathcal{S}^{t+1} = U_{\mathcal{S}}^{t+1}(\mathcal{S}^t, m^{t+1}) = W_2^{\mathcal{S}}(\sigma(W_1^{\mathcal{S}}(\texttt{LayerNorm}(\mathcal{S}^t + m^{t+1})))) \tag{14}$$

The supervirtual node has more expressive power than simply summing or averaging the atom states. The final molecular representations are the learned graph-level vectors that encode structural information about the molecular graph and chemical information including the functional groups, followed by a task-dependent feed-forward neural network for prediction.

3 Experiments

In this section, we present the details of our experiments, including the datasets used, the strategies to train our models and the performance on different benchmarks.

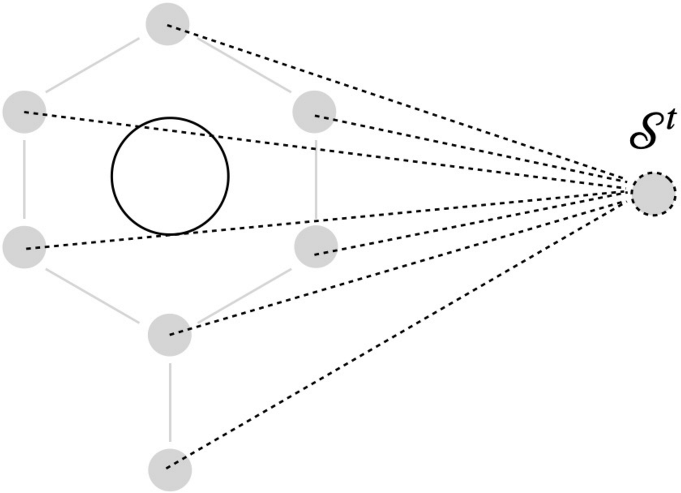
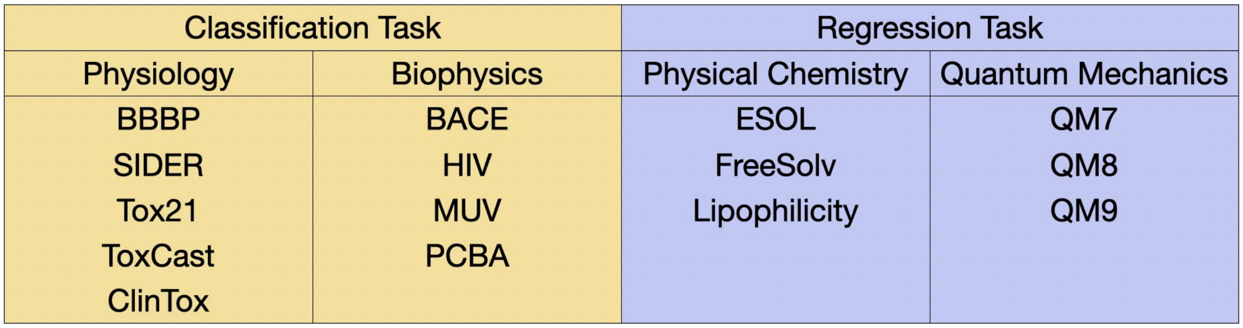


Figure 4: Supervirtual node $\mathcal{S}^t$ is connected to all atoms to update the molecular representations

3.1 Datasets and Metrics for Downstream Tasks

Our objective is to employ D-GATs to process molecular graphs and predict molecular properties, which can be either classification tasks (e.g., predicting the toxicity of compounds) or regression tasks (e.g., predicting the free energy of molecules). To achieve this goal, we have selected 15 benchmark datasets (see Table 2) from MoleculeNet [30], including physiology tasks (BBBP, SIDER, Tox21, ToxCast, ClinTox), biophysics tasks(BACE, HIV, MUV, PCBA), physical chemistry tasks(ESOL, FreeSolv, Lipo) and quantum mechanics tasks (QM7, QM8, QM9).



Classification Task		Regression Task	
Physiology	Biophysics	Physical Chemistry	Quantum Mechanics
BBBP	BACE	ESOL	QM7
SIDER	HIV	FreeSolv	QM8
Tox21	MUV	Lipophilicity	QM9
ToxCast	PCBA		
ClinTox			

Table 2: Datasets used for downstream tasks.

The physiology tasks and biophysics tasks are classification tasks, related to drug design or government agencies' decision-making processes to identify the environmental chemicals that pose the greatest potential risk to human health. For example, the HIV dataset comprises more than 40000 compounds and the target is their ability to inhibit HIV replication, making it a classification task between inactive and active. The Tox21 dataset aims to help scientists understand the potential of chemicals and compounds that may result in toxic effects to human health, containing qualitative toxicity measurements for 8014 compounds on 12 different targets, including nuclear receptors and stress response pathways.

Solubility (ESOL), solvation free energy (FreeSolv) and lipophilicity (Lipo) are fundamental physical chemistry properties that are crucial for understanding how molecules interact with solvents. Quantum mechanics tasks involve predicting geometric, energetic, electronic, and thermodynamic properties (e.g., atomization energy, HOMO/LUMO eigenvalues, etc.), which are typically calculated through solving Schrödinger's equation (approximately using techniques such as *ab* initio density functional theory). These types of prediction tasks are all regression tasks.

To increase the challenge for learning algorithms, all the datasets are scaffold split [37], and the ratio of training, validation, and test sets is 8:1:1. Scaffold splitting ensures that molecules with similar scaffolds are not present in both the training and test sets and allows for evaluating the generalization performance of NNs in molecular property prediction tasks, and could help identify potential limitations or biases in the model that may be present when predicting properties of structurally novel molecules.

The measurements in these datasets can be quantitative or qualitative and we adopt different metrics to compare with previous baselines. As recommended by MoleculeNet [30], we use the average ROC-AUC (area under the receiver operating characteristic curve) [38] as the evaluation metric for the classification datasets, where higher values indicate better performance. For the regression tasks, we use root mean square error (RMSE) for ESOL, FreeSolv and Lipo, and mean average error (MAE) for QM7, QM8 and QM9. The metrics for each dataset are noted in Table 3 and 4.

3.2 Pre-Training

Since the majority of the 15 datasets have been used for testing contain only thousands of molecules, there is a high risk of overfitting, which can lead to a decline in model performance on test set. To mitigate this issue, we employe pre-training and fine-tuning strategy, which offers several benefits such as improved generalization, faster convergence, and better understanding of molecular structures. Pre-training is a form of transfer learning. In pre-training, a model is first trained on a large dataset or a related task, and then fine-tuned on smaller or more specific datasets or tasks. The pre-training step allows the model to learn general features that can be transferred to the downstream tasks, which can help extract high-level features from raw molecular graphs and reduce the amount of training data needed.

The molecular pre-training dataset is based on all public datasets used to verify our model plus the ZINC-250K dataset [29]. For the benchmark datasets, we have restricted our data collection to molecules containing between 10 and 60 atoms, in order to provide our model with a more focused understanding of the intrinsic structure within molecular graphs.

The self-supervised task is extremely important for effective learning from unlabeled data and improving the model's understanding of possible molecular structures. Encouraged by BERT [15], we mask some of the input features and force the model to recover the masked information. More precisely, we mask 16% of the atom features and randomly generated 4% of the atom features and all the bond features connected to these atoms are masked. The goal is to recover the correct atom features.

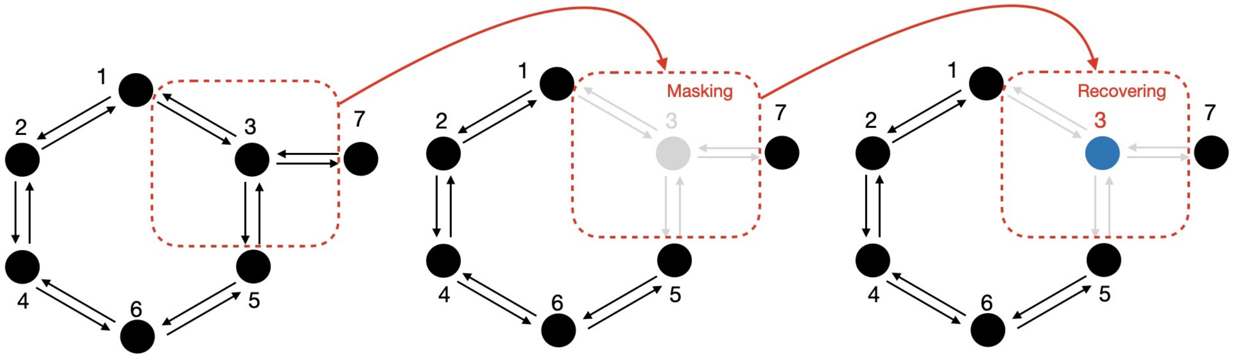


Figure 5: In pre-training stage, some atoms and connected bonds will be masked. The task is only to recover the masked atom features.

According to our message passing algorithm presented in section 2.2, the atom states are updated by directed bond states but independent to the update of bond states. Therefore, the successful recovery of atom features relies on the ability to correctly recover the masked bond features. Hence, we only need to recover atom features in the pre-training stage (see Figure 5).

Nevertheless, the pre-training task to recover masked atom features is an atom-level task. As shown in Figure 2(a), Atom-level tasks allow to train the parameters for directed bond states and for atom states while they do not refer to the readout function. Therefore, we also need the graph-level tasks to pre-train the parameters for molecular representations. Considering that the targets of benchmark datasets should not appeared in pre-training task, only the molecular properties in ZINC-250K dataset are used for pre-training tasks. Thus there are three pre-training regression tasks for: logP (water–octanal partition coefficient), SAS (synthetic accessibility score) and QED (Qualitative Estimate of Drug-likeness).

Pre-training model is composed of the stacked D-GATs (for extracting features for bonds, atoms and molecules) and the feed-forward NNs (for converting representations from D-GATs into atom features or molecular properties) for pre-training tasks. For different downstream tasks shown in Table 2, with parameters in pre-trained D-GATs being slightly optimized, only a single layer feed-forward NNs for fine-tuning tasks need to be trained to transform molecular representations into molecular properties.

For our model, we set four interaction layers in D-GATs, with the dimension of model D_h set to 512, the dropout rate of 0.1, and the number of heads for multi-head attention mechanism of 8.

Classification Tasks ROC-AUC% (higher is better)

Dataset	BBBP	SIDER	Tox21	ToxCast	ClinTox	BACE	HIV	MUV	PCBA	Avg
Number of molecules	**2039**	**1427**	**7831**	**8575**	**1478**	**1513**	**41127**	**93087**	**437929**	
Number of prediction tasks	**1**	**27**	**12**	**617**	**2**	**1**	**1**	**17**	**128**	
GROVER$_{large}$	67.8(0.2)	62.2(1.9)	73.5(0.1)	65.3(0.1)	78.8(0.7)	81.4(1.3)	72.8(0.5)	67.8(1.3)	83.1(0.5)	72.5
AttentiveFP	65.2(0.9)	60.7(1.6)	76.7(0.5)	67.4(0.1)	82.8(0.6)	80.9(0.3)	75.4(0.9)	73.0(0.5)	80.3(0.6)	73.6
D-MPNN	71.2(0.3)	60.2(0.4)	75.1(0.2)	64.3(0.4)	89.6(0.2)	80.0(0.2)	76.4(1.4)	75.3(1.8)	86.2(0.3)	75.4
Pretrain-GCN	70.5(0.9)	62.5(0.2)	75.8(0.3)	65.4(0.1)	63.5(1.8)	84.1(0.4)	76.9(0.7)	79.6(0.2)	84.7(0.1)	73.7
Pretrain-GIN	70.4(0.3)	62.9(0.1)	78.1(0.5)	65.7(0.5)	73.1(1.2)	84.4(0.2)	79.6(0.1)	82.0(0.1)	**86.5(0.1)**	75.9
GEM	71.6(1.3)	60.6(1.0)	77.4(0.7)	67.5(0.5)	89.3(0.2)	82.8(1.2)	78.0(0.8)	74.7(0.7)	86.3(0.4)	76.5
D-GATs	**71.7(0.2)**	**65.8(0.6)**	**78.6(0.2)**	**67.7(0.2)**	**90.9(0.7)**	**84.5(0.3)**	**79.8(0.1)**	**82.5(0.6)**	85.6(0.1)	**78.6**

Table 3: Comparison of performance for molecular property prediction classification tasks

3.3 Results

We compare D-GATs with multiple baselines, including supervised and pretraining baselines. D-MPNN [27] and AttentiveFP [35] are supervised GNNs methods. GROVER [8], PretrainGNN [24] and GEM [9] are pre-training methods.

As suggested by the MoleculeNet [30], the mean and standard deviation of the results for three random seeds are listed in Table 3 and Table 4. The best results are marked in bold.

Our results suggest the following trends:

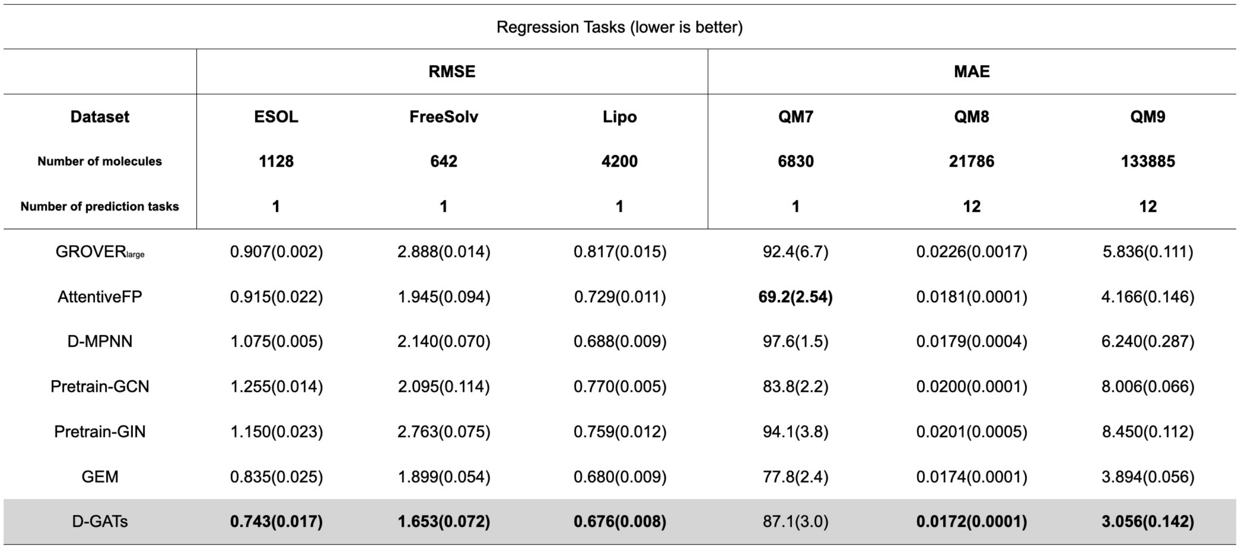
1) Overall, D-GATs outperformed baselines on 13 out of 15 downstream datasets. And on some datasets (e.g., ClinTox and FreeSolv), D-GATs achieved an impressive improvement. Specifically, in the classification datasets, we saw from Table 3 that D-GATs gave the most promising performance, leading to an increase in average ROC-AUC of 2.1% over the previous SOTA results.

2) For D-GATs, the simple pre-training strategy, recovering the masked atom inputs and supervised learning for ReadOut part, was enough to discover the intrinsic rules of molecules. Besides, the pre-training model was successfully generalized to large molecules which had more atoms than those appearing in the pre-training stage.

3) Nevertheless, D-GATs failed to beat SOTA result on the QM7 datasets (see Table 4) due to issues with overfitting. As for the PCBA dataset, the imbalanced samples as well as unlabelled data had a significant negative impact on the model's performance.

4 Conclusion

After extensive evaluation, our findings demonstrate that the message passing algorithm in D-GATs yields superior performance in learning molecular representations compared to



Regression Tasks (lower is better)						
	RMSE			MAE		
Dataset	**ESOL**	**FreeSolv**	**Lipo**	**QM7**	**QM8**	**QM9**
Number of molecules	**1128**	**642**	**4200**	**6830**	**21786**	**133885**
Number of prediction tasks	**1**	**1**	**1**	**1**	**12**	**12**
GROVER$_{large}$	0.907(0.002)	2.888(0.014)	0.817(0.015)	92.4(6.7)	0.0226(0.0017)	5.836(0.111)
AttentiveFP	0.915(0.022)	1.945(0.094)	0.729(0.011)	**69.2(2.54)**	0.0181(0.0001)	4.166(0.146)
D-MPNN	1.075(0.005)	2.140(0.070)	0.688(0.009)	97.6(1.5)	0.0179(0.0004)	6.240(0.287)
Pretrain-GCN	1.255(0.014)	2.095(0.114)	0.770(0.005)	83.8(2.2)	0.0200(0.0001)	8.006(0.066)
Pretrain-GIN	1.150(0.023)	2.763(0.075)	0.759(0.012)	94.1(3.8)	0.0201(0.0005)	8.450(0.112)
GEM	0.835(0.025)	1.899(0.054)	0.680(0.009)	77.8(2.4)	0.0174(0.0001)	3.894(0.056)
D-GATs	**0.743(0.017)**	**1.653(0.072)**	**0.676(0.008)**	87.1(3.0)	**0.0172(0.0001)**	**3.056(0.142)**

Table 4: Comparison of performance for molecular property prediction regression tasks

other GNNs. Notably, D-GATs achieves this with only the most basic features of atoms and bonds, yet outperforms several strong baseline models on both classification and regression tasks. D-GATs comprises three key components: an attention-based scheme to update bond and atom representations, a readout function for extracting the graph-level representation, and a linear classifier for downstream tasks. Our results highlight the potential of D-GATs as a powerful tool for molecular property prediction tasks.

D-GATs is specifically designed for small size graphs, like the one encountered in most reasonable molecular properties. Although it is less efficient than undirected graph models in term of computational time (approximately three-times slower), the extra computational cost is acceptable. However, as explained in section , the presence of rings in graphs may disrupt the directed message flow. To avoid this problem, the depth of model must be carefully decided. These two limitations make the D-GATs specifically suitable for molecular graphs, but not for large or dense graphs, such as social networks.

In addition, our results demonstrate that the directed bonds in D-GATs outperforms D-MPNN [27] due to the attention mechanism. Our model follows the common MPNN framework and does not require complex operations, with a model in size of approximately 100 MB.

An important future direction of our work is to develop an appropriate pre-training strategy to enhance the generalization ability of D-GATs. In this paper, we followed the strategy presented in BERT [15], masking part of the atom features and surrounded bond features and expecting the pre-training model to recover the masked information. However, more advanced and intricate pre-training strategies exist, such as Context Prediction [24] which allows the model to match the chemical environment, or the geometry-enhanced learning strategy proposed in [9], which leverages 3D information such as bond lengths and angles. Another possible direction is to design better message passing algorithm. For now, the message flows in connected bonds, higher body order messages [39] could merge information with impressive efficiency, improving models' expressive ability without adding more layers.

Acknowledgments and Disclosure of Funding

We would like to thank Jean-Philip Piquemal, Theo Jaffrelot Inizan and Louis Lagardère for helpful discussions. We acknowledge the funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Program (Grant Agreement No. 810367), project EMC2(JPP, YM)

References

[1] Antonio Lavecchia. Machine-learning approaches in drug discovery: methods and applications. *Drug discovery today*, 20(3):318–331, 2015.

[2] Justin S Smith, Adrian E Roitberg, and Olexandr Isayev. Transforming computational drug discovery with machine learning and ai, 2018.

[3] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

[4] Jing Wei, Xuan Chu, Xiang-Yu Sun, Kun Xu, Hui-Xiong Deng, Jigen Chen, Zhongming Wei, and Ming Lei. Machine learning in materials science. *InfoMat*, 1(3):338–358, 2019.

[5] Qingda Zang, Kamel Mansouri, Antony J Williams, Richard S Judson, David G Allen, Warren M Casey, and Nicole C Kleinstreuer. In silico prediction of physicochemical properties of environmental chemicals using molecular fingerprints and machine learning. *Journal of chemical information and modeling*, 57(1):36–49, 2017.

[6] Minjian Yang, Bingzhong Tao, Chengjuan Chen, Wenqiang Jia, Shaolei Sun, Tiantai Zhang, and Xiaojian Wang. Machine learning models based on molecular fingerprints and an extreme gradient boosting method lead to the discovery of jak2 inhibitors. *Journal of Chemical Information and Modeling*, 59(12):5002–5012, 2019.

[7] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[8] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems*, 33:12559–12571, 2020.

[9] Xiaomin Fang, Lihang Liu, Jieqiong Lei, Donglong He, Shanzhuo Zhang, Jingbo Zhou, Fan Wang, Hua Wu, and Haifeng Wang. Geometry-enhanced molecular representation learning for property prediction. *Nature Machine Intelligence*, 4(2):127–134, 2022.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[13] Zheng Xu, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery. In *Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics*, pages 285–294, 2017.

[14] Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, pages 429–436, 2019.

[15] Josh Payne, Mario Srouji, Dian Ang Yap, and Vineet Kosaraju. Bert learns (and teaches) chemistry. *arXiv preprint arXiv:2007.16012*, 2020.

[16] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[17] Alessandro Sperduti and Antonina Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[18] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, volume 2, pages 729–734, 2005.

[19] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[20] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[23] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[24] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[25] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

[26] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International conference on machine learning*, pages 2323–2332. PMLR, 2018.

[27] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.

[28] Katsuhiko Ishiguro, Shin-ichi Maeda, and Masanori Koyama. Graph warp module: an auxiliary module for boosting the power of graph neural networks in molecular graph analysis. *arXiv preprint arXiv:1902.01020*, 2019.

[29] John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005.

[30] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[31] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the oversmoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3438–3445, 2020.

[32] Xu Han, Ming Jia, Yachao Chang, Yaopeng Li, and Shaohua Wu. Directed message passing neural network (d-mpnn) with graph edge attention (gea) for property prediction of biofuel-relevant species. *Energy and AI*, 10:100201, 2022.

[33] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015.

[34] Chen Qian, Yunhai Xiong, and Xiang Chen. Directed graph attention neural network utilizing 3d coordinates for molecular property prediction. *Computational Materials Science*, 200:110761, 2021.

[35] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16):8749–8760, 2019.

[36] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[37] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[38] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[39] Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 35:11423–11436, 2022.