Bioinformatics, 38(7), 2022, 2003–2009
https://doi.org/10.1093/bioinformatics/btac039
Advance Access Publication Date: 30 January 2022
Original Paper

Data and text mining

Cross-dependent graph neural networks for molecular property prediction

Hehuan Ma [1], Yatao Bian[2,†], Yu Rong[2,†], Wenbing Huang[3], Tingyang Xu[2], Weiyang Xie[2], Geyan Ye[2] and Junzhou Huang[1,*]

[1]Department of Computer Science, University of Texas at Arlington, Arlington 76019, USA, [2]AI Lab, Tencent, Shenzhen 518057, China and [3]Institute for AI Industry Research, Tsinghua University, Beijing 100084, China

*To whom correspondence should be addressed.
†The authors wish it to be known that these authors contributed equally.

Associate Editor: Zhiyong Lu

Received on October 1, 2021; revised on December 14, 2021; editorial decision on January 11, 2022

Abstract

Motivation: The crux of molecular property prediction is to generate meaningful representations of the molecules. One promising route is to exploit the molecular graph structure through graph neural networks (GNNs). Both atoms and bonds significantly affect the chemical properties of a molecule, so an expressive model ought to exploit both node (atom) and edge (bond) information simultaneously. Inspired by this observation, we explore the multi-view modeling with GNN (MVGNN) to form a novel paralleled framework, which considers both atoms and bonds equally important when learning molecular representations. In specific, one view is atom-central and the other view is bond-central, then the two views are circulated via specifically designed components to enable more accurate predictions. To further enhance the expressive power of MVGNN, we propose a cross-dependent message-passing scheme to enhance information communication of different views. The overall framework is termed as CD-MVGNN.

Results: We theoretically justify the expressiveness of the proposed model in terms of distinguishing non-isomorphism graphs. Extensive experiments demonstrate that CD-MVGNN achieves remarkably superior performance over the state-of-the-art models on various challenging benchmarks. Meanwhile, visualization results of the node importance are consistent with prior knowledge, which confirms the interpretability power of CD-MVGNN.

Availability and implementation: The code and data underlying this work are available in GitHub at https://github.com/uta-smile/CD-MVGNN.

Contact: jzhuang@uta.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Molecular property prediction is a fundamental but challenging task in drug discovery, and has attracted increasing attention in the last decades with the developments of deep-learning techniques. A molecule could be represented as a SMILES sequence, or a topological graph by treating atoms as nodes, and chemical bonds as edges. Thus, sequence-based and graph-based deep-learning methods become superior to exploring the molecular representation (Gilmer *et al.*, 2017; Guo and Wu, 2020; Wang *et al.*, 2019b; Xu *et al.*, 2017; Yang *et al.*, 2019). To date, graph neural networks (GNNs) have gained more popularity due to their capability of modeling graph-structured data. Successes have been achieved in various domains, such as social network (Bian *et al.*, 2020; Chang *et al.*, 2021; Li *et al.*, 2019; Veličković *et al.*, 2017; Zhao *et al.*, 2021), computer vision (Zeng *et al.*, 2019) and bioinformatics (Raju *et al.*, 2020; Wang *et al.*, 2019a; Yan *et al.*, 2020; Yang *et al.*, 2021). Molecular property prediction is also a promising application of GNNs. In this sense, a molecular property prediction task is equivalent to a supervised graph classification problem [see, e.g. toxicity prediction (Pires *et al.*, 2015; Rong *et al.*, 2020)].

Despite the fruitful results obtained by GNNs, there remain three limitations: (i) most of the GNN models focus either on node-oriented or edge-oriented. However, nodes and edges play equally important roles in many practical scenarios. Specifically, molecules with different atoms (nodes) but same bonds (edges) are distinct compounds with different properties, and so as to different bonds (edges) but same atoms (nodes). For example, as shown in Figure 1 (upper), equipped with the same bonds, only one-atom difference make the two molecules distinct Octanol/Water Partition Coefficients. Caffeine is more hydrophilic while 6-thiocaffeine is more lipophilic (Bhal, 2007). Similarly, in Figure 1 (lower), the molecular formulas of acetone and propen-2-ol are exactly the same,

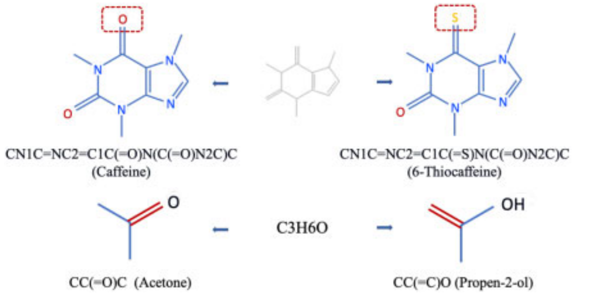

Fig. 1. The upper two molecules share the same bond structures, but contain different atoms. The lower two molecules share the same atoms, but equip with different bonds

but the bond difference makes acetone behave mild irritation to human eyes, nose, skin, etc. (Kim *et al.*, 2016). Accordingly, both nodes and edges are fairly essential for molecular property prediction. Therefore, how to properly integrate *both node and edge information in a unified manner* is the first challenge. (ii) As proved in Xu *et al.* (2018), the GNN model with message-passing scheme is at most as strong as the WL graph isomorphism test (Weisfeiler and Lehman, 1968), which limits the expressive power of GNNs and harms the performance of down-stream tasks. (iii) Existing GNNs usually lack interpretability power, which is actually crucial for drug discovery tasks. Take molecular property prediction as an example; being aware of how the model validates the property will help practitioners figure out the key components that determine specific properties (Preuer *et al.*, 2019).

In pursuit of tackling the above challenges, we explore the idea of multi-view learning (Zhao *et al.*, 2017), and develop a new form of GNN: Multi-View Graph Neural Network (MVGNN), which contains two sub-modules that generate the graph embeddings from node and edge, respectively. Therefore, it investigates the molecular graph from two views simultaneously. Furthermore, we design a cross-dependent message-passing scheme to break the expressive power barrier of MVGNN and propose cross-dependent MVGNN (CD-MVGNN). We theoretically justify that the expressiveness of CD-MVGNN is strictly more powerful than Graph Isomorphisom Network (Xu *et al.*, 2018) in terms of distinguishing non-isomorphism graphs. Lastly, we employ a shared self-attentive aggregation module to produce the graph-level embeddings and interpretability results, and a disagreement loss to stabilize the training process of the multi-view pipeline. Comprehensive experiments on 11 benchmarks demonstrate the superiority of proposed CD-MVGNN model. Namely, the overall performance of CD-MVGNN and MVGNNZMVGNN achieve up to 3.62% improvement on classification benchmarks and 23.55% improvement on regression benchmarks compared with state-of-the-art (SOTA) methods. Moreover, case studies on toxicity prediction demonstrate the interpretability power of proposed model.

2 Materials and methods

2.1 Preliminaries on molecular representations and generalized GNNs

We abstract a molecule c as a topological graph $G_c = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = p$ refers to the set of p nodes (atoms) and $|\mathcal{E}| = q$ refers to a set of q edges (bonds). $\mathcal{N}_v$ denotes the neighborhood set of node v. We denote the feature of node v as $\mathbf{x}_v \in \mathbb{R}^{d_n}$ and the feature of edge (v, k) as $\mathbf{e}_{vk} \in \mathbb{R}^{d_e}$ (with a bit abuse of notations, $\mathbf{e}_{vk}$ can represent either the edge (v, k) or the edge features). d_n and d_e refer to the feature dimensions of nodes and edges, respectively. Exemplar node and edge features are the chemical relevant features, such as atomic mass and bond type. Please refer to Supplementary Section 2.5 for detailed feature extraction process. Properties of a molecule $\mathbf{y}$ constitute the targets of the predictive task. Given a molecule c and its associated graph representation G_c, molecular property prediction aims to predict the properties $\mathbf{y}_c$ according to the embedding ξ_c of G_c. The values of $\mathbf{y}$ are either categorical values [e.g. toxicity (Richard *et al.*, 2016)] for classification tasks or real values [e.g. atomization energy and the electronic spectra (Ramakrishnan *et al.*, 2015)] for regression tasks.

Generalized GNNs. Most of the GNN models are built upon the message-passing process, which aggregates and passes the feature information of corresponding neighboring nodes to produce new hidden states of the nodes. After the message-passing process, all hidden states of the nodes are fed into a readout component, to produce the final graph-level embedding. Here, we present a generalized version of the message-passing scheme. Suppose there are L iterations/layers, and iteration l contains K_l hops. In iteration l, the k-th hop of message passing can be formulated as,

$$\begin{aligned} \mathbf{m}_v^{(l,k)} &= \mathrm{AGG}^{(l)}(\{\mathbf{h}_v^{(l,k-1)}, \mathbf{h}_u^{(l,k-1)}, \mathbf{e}_{uv} \mid u \in \mathcal{N}_v\}), \\ \mathbf{h}_v^{(l,k)} &= \mathrm{MLP}^{(l)}(\mathbf{m}_v^{(l,k)}), \end{aligned} \tag{1}$$

where we make the convention that $\mathbf{h}_v^{(l,0)} := \mathbf{h}_v^{(l-1,K_{l-1})}$. $\mathrm{AGG}^{(l)}$ denotes the aggregation function, $\mathbf{m}_v^{(l,k)}$ is the aggregated message and $\mathrm{MLP}^{(l)}$ is a multi-layer perceptron (for instance, it could be a one layer neural net, then the state update becomes $\mathbf{h}_v^{(l,k)} = \sigma(\mathbf{W}^{(l)}\mathbf{m}_v^{(l,k)} + \mathbf{b}^{(l)})$, where σ stands for the activation function). There are several popular choices for the aggregation function $\mathrm{AGG}^{(l)}$, such as mean, max pooling and the graph attention mechanism (Veličković *et al.*, 2017). Note that for one iteration of message passing, there are a layer of trainable parameters (parameters inside $\mathrm{AGG}^{(l)}$ and $\mathrm{MLP}^{(l)}$). These parameters are shared across the K_l hops within iteration l. After L iterations of message passing, the hidden states of the last hop in the last iteration are used as the embeddings of the nodes, i.e. $\mathbf{h}_v^{(L,K_L)}, v \in \mathcal{V}$. Lastly, a READOUT operation is applied to generate the graph-level representation,

$$\mathbf{h}_G = \mathrm{READOUT}(\{\mathbf{h}_v^{(0,K_0)}, \ldots, \mathbf{h}_v^{(L,K_L)} \mid v \in \mathcal{V}\}). \tag{2}$$

If choosing the sum aggregation with a learnable parameter $\epsilon^{(l)}$, i.e. $\mathrm{AGG}^{(l)}(\{\mathbf{h}_v^{(l,k-1)}, \mathbf{h}_u^{(l,k-1)}, \mathbf{e}_{uv} | u \in \mathcal{N}_v\}) = ((1 + \epsilon^{(l)})\mathbf{h}_v^{(l,k-1)} + \sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(l,k-1)}) || (\sum_{u \in \mathcal{N}_v} \mathbf{e}_{uv})$ ($||$ is the concatenation operation), then generalized GNN recovers graph isomorphism network architecture (Xu *et al.*, 2018), which provably generalizes the WL graph isomorphism test (Weisfeiler and Lehman, 1968).

2.2 Overview of CD-MVGNN

CD-MVGNN is implemented in a multi-view fashion, namely, it equally considers both atom features and bond features for constituting a molecular representation. As shown in Figure 2, the proposed architecture contains two concurrent sub-modules, *Node-central encoder* and *Edge-central encoder*. A *cross-dependent message-passing* scheme is applied between the two encoders to enable messages circulate and update during the training process.

Next, CD-MVGNN adopts an aggregation function to produce the graph embedding vector from the node/edge encoder. Other than the mean-pooling mechanism, we propose to use the *self-attentive aggregation* to learn different weights of the node/edge embeddings to produce the final graph embedding. Furthermore, the self-attentive aggregation layer is shared between the node-central and edge-central encoders, to reinforce the learning of the node features and the edge features, respectively. After the self-attentive aggregation, CD-MVGNN feeds the corresponding graph embeddings to two MLPs to fit the loss function. To stabilize the training process of this multi-view architecture, we employ the *disagreement loss* to enforce the outputs of the two MLPs to be close with each other.

2.3 Node-central and edge-central encoders

To ease the exposition, in the sequel when using one single superscript, we mean the hop index k while ignoring the layer/iteration index l.

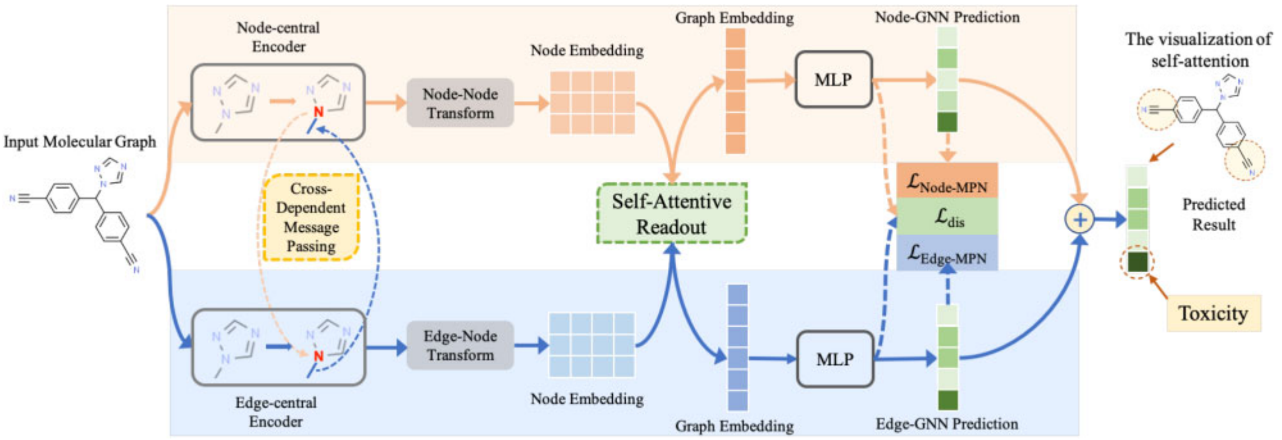

Fig. 2. Overview of CD-MVGNN model. CD-MVGNN model passes the graph through two encoders to generate two sets of node embeddings. A cross-dependent message-passing scheme is applied between two encoders to ensure the information flow circulation. A shared self-attention readout learns the node importance and produces two graph embeddings accordingly. The embeddings are then fed into two MLPs to make predictions. The final prediction is the ensemble of the two predictions. Furthermore, by visualizing the learned attentions over nodes, one can identify the atoms/functional groups that are responsible for the predictions. For example, CD-MVGNN finds out that the cyano groups contribute to the toxicity significantly

Node-central encoder.

Node-GNN is built upon the generalized message passing in Equation (1). Additionally, we add *input* and *output* layers, to enhance its expressive power. Specifically,

$$\begin{aligned}\mathbf{m}_v^{(k)} &= \text{AGG}_{\text{node}}(\{\mathbf{h}_v^{(k-1)}, \mathbf{h}_u^{(k-1)}, \mathbf{e}_{uv} \mid u \in \mathcal{N}_v\}),\\ \mathbf{h}_v^{(k)} &= \text{MLP}_{\text{node}}(\{\mathbf{m}_v^{(k)}, \mathbf{h}_v^{(0)}\}),\end{aligned} \tag{3}$$

where $\mathbf{h}_v^{(0)} = \sigma(\mathbf{W}_{\text{nin}}\mathbf{x}_v)$ is the input state of Node-GNN, $\mathbf{W}_{\text{nin}} \in \mathbb{R}^{d_{\text{hid}} \times d_n}$ is the input weight matrix. The input layer can also be viewed as a residual connection.

After L iterations of message passing, we utilize an additional message-passing step with a new weight matrix $\mathbf{W}_{\text{nout}} \in \mathbb{R}^{d_{\text{out}} \times (d_n + d_{\text{hid}})}$ to produce the final node embeddings:

$$\begin{aligned}\mathbf{m}_v^{\text{o}} &= \text{AGG}_{\text{node}}(\{\mathbf{h}_v^{(L,K_L)}, \mathbf{h}_u^{(L,K_L)}, \mathbf{x}_u | u \in \mathcal{N}_v\}),\\ \mathbf{h}_v^{\text{o}} &= \sigma(\mathbf{W}_{\text{nout}}\mathbf{m}_v^{\text{o}}).\end{aligned} \tag{4}$$

We denote $\mathbf{H}_n = [\mathbf{h}_1^{\text{o}}, \dots, \mathbf{h}_p^{\text{o}}] \in \mathbb{R}^{d_{\text{out}} \times p}$ as the output embeddings of Node-GNN, where d_{out} is the dimension of the output embeddings.?? in Supplementary Section 2.1 illustrates the process of message-passing process in Node-GNN.

Edge-central encoder. In classical graph theory, the line graph $L(G)$ of a graph G is the graph that encodes the adjacencies between edges of G (Harary and Norman, 1960). $L(G)$ provides a fresh perspective to understand the original graph, i.e. the nodes are viewed as the connections while edges are viewed as entities. Therefore, it enables to perform message-passing operation through edges to imitate Node-GNN on $L(G)$ (Yang *et al.*, 2019). Namely, given an edge (v, w), we can formulate the edge-based GNN (Edge-GNN) as:

$$\begin{aligned}\mathbf{m}_{vw}^{(k)} &= \text{AGG}_{\text{edge}}(\{\mathbf{h}_{vw}^{(k-1)}, \mathbf{h}_{uv}^{(k-1)}, \mathbf{x}_u | u \in \mathcal{N}_v \setminus w\}),\\ \mathbf{h}_{vw}^{(k)} &= \text{MLP}_{\text{edge}}(\{\mathbf{m}_{vw}^{(k-1)}, \mathbf{h}_{vw}^{(0)}\}),\end{aligned} \tag{5}$$

where $\mathbf{h}_{vw}^{(0)} = \sigma(\mathbf{W}_{\text{ein}}\mathbf{e}_{vw})$ is the input state of Edge-GNN, $\mathbf{W}_{\text{ein}} \in \mathbb{R}^{d_{\text{hid}} \times d_e}$ is the input weight matrix. In Equation (5), the state vector is defined on edge $\mathbf{e}_{vw}$ and the neighboring edge set of $\mathbf{e}_{vw}$ is defined as all edges connected to the start node v except the node w.?? shows an example of the message-passing process in Edge-GNN.

After recurring L steps of message passing, the output of Edge-GNN is the state vectors for edges. In order to incorporate the shared-attentive readout to generate the graph embedding, one more round of message passing on nodes is employed to transform edge-wise embeddings to node-wise embeddings, and generate the second set of node embeddings. Specifically, the edge–node transform is established by the following,

$$\begin{aligned}\mathbf{m}_v^{\text{o}} &= \text{AGG}_{\text{edge}}(\{\mathbf{h}_{vw}^{(L,K_L)}, \mathbf{h}_{uv}^{(L,K_L)}, \mathbf{x}_u | u \in \mathcal{N}_v\}),\\ \mathbf{h}_v^{\text{o}} &= \sigma(\mathbf{W}_{\text{eout}}\mathbf{m}_v^{\text{o}}),\end{aligned} \tag{6}$$

where $\mathbf{W}_{\text{eout}} \in \mathbb{R}^{d_{\text{out}} \times (d_n + d_{\text{hid}})}$ specifies the weight matrix. Therefore, the final output of Edge-GNN provides a new set of *node* embeddings from the edge message-passing process. This set of node embeddings are denoted as $\mathbf{H}_e = [\mathbf{h}_1^{\text{o}}, \dots, \mathbf{h}_p^{\text{o}}] \in \mathbb{R}^{d_{\text{out}} \times p}$.

2.4 Cross-dependent message-passing scheme

After constructing the MVGNN framework, we notice that the information flow is not sufficiently efficient, even though the MVGNN model has been proved to have superior performance for many molecular property prediction tasks (as verified in the experiments).

Suppose all the information needed to predict the property resides in the molecule itself. For MVGNN, the information flows through two distinct paths in parallel: one path is the node-central encoder, and the other one is the edge-central encoder. The information from the two paths finally joins at the disagreement loss.

However, the two flows of information could meet *earlier*, to enable more efficient information communication. In pursuit of this point, we propose the *cross-dependent message-passing* scheme. On a high level, it makes the message-passing operations of the node and edge cross-dependent with each other. Specifically, we change the message-passing operations of the node and edge encoders [in Equations (3) and (5), respectively] to be:

$$\begin{aligned}\mathbf{m}_v^{(k)} &= \text{AGG}_{\text{node}}(\{\mathbf{h}_v^{(k-1)}, \mathbf{h}_u^{(k-1)}, \mathbf{h}_{vu}^{(k-1)}, \mathbf{e}_{vu} | u \in \mathcal{N}_v\}),\\ \mathbf{h}_v^{(k)} &= \text{MLP}_{\text{node}}(\{\mathbf{m}_v^{(k)}, \mathbf{h}_v^{(0)}\}),\\ \mathbf{m}_{vw}^{(k)} &= \text{AGG}_{\text{edge}}(\{\mathbf{h}_{vw}^{(k-1)}, \mathbf{h}_{uv}^{(k-1)}, \mathbf{h}_u^{(k-1)}, \mathbf{x}_u | u \in \mathcal{N}_v \setminus w\}),\\ \mathbf{h}_{vw}^{(k)} &= \text{MLP}_{\text{edge}}(\{\mathbf{m}_{vw}^{(k)}, \mathbf{h}_{vw}^{(0)}\}).\end{aligned} \tag{7}$$

The first two equations indicate new node message passing, while the other two show edge message passing. One can see that when applying aggregation in node message passing, we use the *newest* hidden states of edges. While conducting aggregation in edge message passing, it requires the newest hidden states of nodes. In this manner, the two paths of information flow become cross-dependent with each other. Figure 3 gives an illustration of this scheme, take the node-view therein e.g. the attached edge features $\mathbf{h}_{vu}^{(k-1)}$ used in the current aggregation phase are obtained from the previous step of message passing in the edge-GNN. A similar approach is performed for the edge-central encoder. Such circulation between the two encoders during every message-passing step ensures the information flow stays updated, empowering CD-MVGNN to be more efficient. We will empirically show that the cross-dependent message-passing scheme enables more expressive power compared to the basic MVGNN

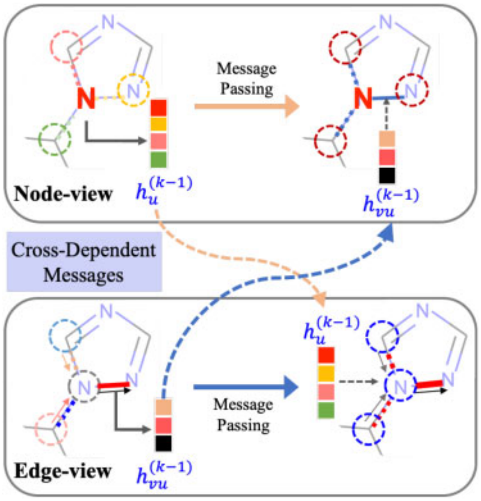

Fig. 3. Illustration of the cross-dependent message-passing scheme. Node message passing utilizes the newest hidden states of neighboring edges, while edge message passing uses the newest hidden states of neighboring nodes. This scheme enables more efficient communication between the two views

architecture. Moreover, a detailed proof that theoretically justifies the expressive power of CD-MVGNN under the framework of Xu *et al.* (2018) is deferred to Supplementary Section 2.4.

2.5 Interpretable readout and disagreement loss

Interpretable readout. Instead of MEAN Readout, We adopt the *interpretable shared self-attentive readout component* (Li *et al.*, 2019) to generate the graph embeddings from the node representations.

Namely, given an output of node-central encoder $\mathbf{H}_n \in \mathbb{R}^{d_{\text{out}} \times p}$, the self-attention $\mathbf{S}$ over nodes is:

$$\mathbf{S} = \text{softmax}\Big(\mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{H}_n)\Big), \quad (8)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{attn}} \times d_{\text{out}}}$ and $\mathbf{W}_2 \in \mathbb{R}^{r \times d_{\text{attn}}}$ are learnable matrices. Given $\mathbf{S}$, we can obtain the graph-level embeddingby $\xi_n = \text{Flatten}(\mathbf{S}\mathbf{H}_n^\top)$. The self-attention $\mathbf{S}$ implies the importance of the nodes when generating graph embedding, hence indicating contributions of the nodes for down-stream tasks, which equips CD-MVGNN with interpretability power. Detail explanation is deferred to Supplementary Section 2.2.

The disagreement loss. For the loss function, we employ an additional mean squared error $\mathcal{L}_{\text{dis}} = \sum_{G_i \in \mathcal{G}} |\gamma_{n,i} - \gamma_{e,i}|^2$ to restrain the predictions from node-central and edge-central encoders, which terms as *disagreement loss*. The overall loss function is formed as: $\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda \mathcal{L}_{\text{dis}}$, where λ is a tradeoff hyperparameter, and $\mathcal{L}_{\text{pred}}$ is the corresponding supervised loss based on the task types, say, cross-entropy for classification and mean squared error for regression. Detailed descriptions are included in Supplementary Section 2.3.

3 Results and discussion

We conduct performance evaluations of MVGNN and CD-MVGNN with various SOTA baselines on molecular property classification and regression tasks. We also perform ablation studies on different components of our models. Lastly, we conduct case studies to demonstrate the interpretability power of the proposed models.

3.1 Experimental setup

Datasets. We experimented with 11 popular benchmark datasets, among which 6 are classification tasks and the others are regression tasks. Specifically, BACE is about the biophysics property; BBBP, Tox21, Toxcast, SIDER and Clintox record several molecular physiology properties; QM7 and QM8 contain molecular quantum mechanics information; ESOL, Lipophilicity and Freesolv document physical chemistry properties (Wu *et al.*, 2018). Details are deferred to Supplementary Section 3.1.

Baselines. We thoroughly evaluate the performance of our methods against popular baselines from both machine learning and chemistry communities. Among them, Influence Relevance Voting (IRV) (Swamidass *et al.*, 2009) and LogReg (Friedman *et al.*, 2000) utilize different traditional machine-learning approaches. GraphConv (Duvenaud *et al.*, 2015), Weave (Kearnes *et al.*, 2016), SchNet (Schütt *et al.*, 2017), MGCN (Lu *et al.*, 2019), N-Gram (Liu *et al.*, 2019), AttentiveFP (Xiong *et al.*, 2020), CMPNN (Song *et al.*, 2020), GIN (Xu *et al.*, 2018), MPNN (Gilmer *et al.*, 2017) and DMPNN (Yang *et al.*, 2019) are GNN-based models. CoMPT and GROVER are two recently proposed transformer-based models. Details can be found in Supplementary Section 3.2.

Dataset splitting and experimental setting. We apply the *scaffold splitting* for all tasks on all datasets, which is more practical and challenging than random splitting. More details about this splitting method are introduced in Supplementary Section 3.1.

Evaluation metrics. We follow the evaluation criteria utilized by the baseline models. In specific, all classification tasks are evaluated by AUC-ROC. For the regression tasks, we apply MAE and RMSE to evaluate the performance on different datasets.

Two naive schemes setup. To demonstrate the effectiveness of the shared self-attentive readout and the disagreement loss in the multi-view architecture, we also implement two naive schemes. Concat + Mean concatenates the mean-pooling outputs of the two sub-modules, and Concat + Attn concatenates the self-attentive outputs (we do not share the attention here) of the two sub-modules.

3.2 Experiment results

Table 1 summarizes the results of the classification tasks. To evaluate the robustness of our method, we report the mean and standard deviation of 10 times runs with different random seeds for MVGNN, CD-MVGNN and the other models. Table 1 indicates the following: (i) our MVGNN and CD-MVGNN models gain significant enhancement against SOTAs on all datasets consistently, CD-MVGNN performs slightly better than MVGNN with a 1.80% average AUC boost compared with the SOTAs on each dataset, which is regarded as the remarkable boost, considering the challenges on these classification benchmarks with scaffold splitting method. (ii) Compared with the SOTAs, MVGNN and CD-MVGNN has much smaller SD, implying that our models are more robust than the baselines. (iii) Compared with the two simple variants, MVGNN and CD-MVGNN demonstrate the superiority both on the performance and robustness. It validates the effectiveness of the multi-view architecture with self-attentive readout and disagreement loss constraints.

We also conducted regression tasks over five benchmark datasets and nine baseline models, which are deferred in Supplementary Section 3.5.

3.3 Ablation studies on key design choices

We conduct ablation studies to validate the impacts of key components in our proposed models.

Cross-dependent message passing. We plot the average number of parameters in the MVGNN and CD-MVGNN models in Figure 4, these are the models with the best performance in each hyperparameter search. It clearly indicates that CD-MVGNN, while enjoying competitive performance, needs *much less* amount of parameters than MVGNN. Specifically, the average number of parameters of MVGNN is *15.26* times of that of CD-MVGNN. This confirms that the *cross-dependent message-passing* scheme can significantly improve the expressive power of the model, by enabling a more efficient information communication scheme in the multi-view architecture.

Self-attentive readout and disagreement loss. We report the results of three datasets with fixed train/valid/test sets to evaluate the impacts in Table 2, which demonstrates the proposed multi-view models overall performs the best on all three datasets.

Table 1. Performance of classification tasks on AUC-ROC (higher is better) with the scaffold split

Method	BACE	BBBP	Tox21	ToxCast	SIDER	ClinTox
IRV	$0.838_{\pm 0.055}$	$0.877_{\pm 0.051}$	$0.699_{\pm 0.055}$	$0.604_{\pm 0.037}$	$0.595_{\pm 0.022}$	$0.741_{\pm 0.069}$
LogReg	$0.844_{\pm 0.040}$	$0.835_{\pm 0.067}$	$0.702_{\pm 0.028}$	$0.613_{\pm 0.033}$	$0.583_{\pm 0.034}$	$0.733_{\pm 0.084}$
GraphConv	$0.854_{\pm 0.011}$	$0.877_{\pm 0.036}$	$0.772_{\pm 0.041}$	$0.650_{\pm 0.025}$	$0.593_{\pm 0.035}$	$0.845_{\pm 0.051}$
Weave	$0.791_{\pm 0.008}$	$0.837_{\pm 0.065}$	$0.741_{\pm 0.044}$	$0.678_{\pm 0.024}$	$0.543_{\pm 0.034}$	$0.823_{\pm 0.023}$
SchNet	$0.750_{\pm 0.033}$	$0.847_{\pm 0.024}$	$0.767_{\pm 0.025}$	$0.679_{\pm 0.021}$	$0.545_{\pm 0.038}$	$0.717_{\pm 0.042}$
MGCN	$0.734_{\pm 0.030}$	$0.850_{\pm 0.064}$	$0.707_{\pm 0.016}$	$0.663_{\pm 0.009}$	$0.552_{\pm 0.018}$	$0.634_{\pm 0.042}$
N-Gram	$0.876_{\pm 0.035}$	$0.912_{\pm 0.013}$	$0.769_{\pm 0.027}$	–[1]	$0.632_{\pm 0.005}$	$0.855_{\pm 0.037}$
AttentiveFP	$0.863_{\pm 0.015}$	$0.908_{\pm 0.050}$	$0.807_{\pm 0.020}$	$0.579_{\pm 0.001}$	$0.605_{\pm 0.060}$	$0.933_{\pm 0.020}$
CMPNN	$0.869_{\pm 0.023}$	$0.929_{\pm 0.025}$	$0.810_{\pm 0.022}$	$0.709_{\pm 0.006}$	$0.617_{\pm 0.016}$	$0.922_{\pm 0.017}$
GIN	$0.845_{\pm 0.040}$	$0.894_{\pm 0.011}$	$0.811_{\pm 0.028}$	$0.703_{\pm 0.006}$	$0.591_{\pm 0.039}$	$0.869_{\pm 0.076}$
MPNN	$0.815_{\pm 0.044}$	$0.913_{\pm 0.041}$	$0.808_{\pm 0.024}$	$0.691_{\pm 0.013}$	$0.595_{\pm 0.030}$	$0.879_{\pm 0.054}$
DMPNN	$0.852_{\pm 0.053}$	$0.919_{\pm 0.030}$	$0.826_{\pm 0.023}$	$0.718_{\pm 0.011}$	$0.632_{\pm 0.023}$	$0.897_{\pm 0.040}$
CoMPT	$0.838_{\pm 0.035}$	$0.926\ _{\pm 0.028}$	$0.792_{\pm 0.020}$	$0.704_{\pm 0.007}$	$0.612_{\pm 0.026}$	$0.876_{\pm 0.031}$
GROVER[2]	$0.851_{\pm 0.039}$	$0.913\ _{\pm 0.023}$	$0.811_{\pm 0.022}$	$0.713_{\pm 0.010}$	$0.614_{\pm 0.027}$	$0.823_{\pm 0.060}$
Concat + Mean	$0.842_{\pm 0.004}$	$0.930_{\pm 0.002}$	$0.816_{\pm 0.003}$	$0.721_{\pm 0.001}$	$0.621_{\pm 0.007}$	$0.882_{\pm 0.008}$
Concat + Attn	$0.832_{\pm 0.007}$	$0.931_{\pm 0.006}$	$0.819_{\pm 0.003}$	$0.728_{\pm 0.002}$	$0.632_{\pm 0.008}$	$0.913_{\pm 0.009}$
MVGNN	$0.863_{\pm 0.002}$	$\mathbf{0.938}_{\pm 0.003}$	$0.833_{\pm 0.001}$	$0.729_{\pm 0.006}$	$\mathbf{0.644}_{\pm 0.003}$	$0.930_{\pm 0.003}$
CD-MVGNN	$\mathbf{0.892}_{\pm 0.011}$	$0.933_{\pm 0.006}$	$\mathbf{0.836}_{\pm 0.006}$	$\mathbf{0.744}_{\pm 0.005}$	$0.639_{\pm 0.012}$	$\mathbf{0.945}_{\pm 0.017}$

Note: Best score is marked as bold, and the last two rows indicate the results of our methods.

[1] Result not presented since N-Gram requires task-based preprocessing, which cannot be finished in reasonable days.

[2] GROVER is utilized without pretrained models for a fair comparison.

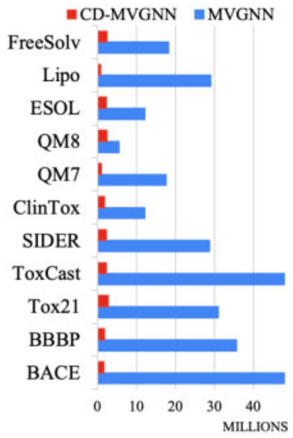

Fig. 4. Model parameters comparison

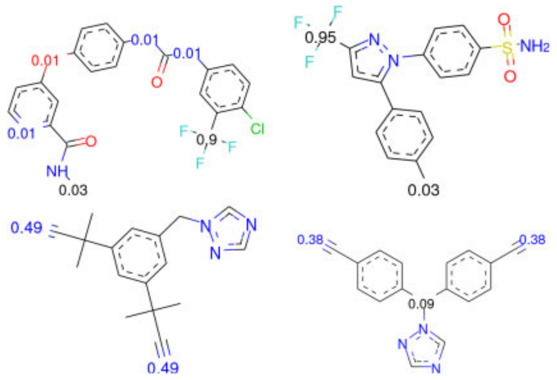

Fig. 5. Visualization of attention values on ClinTox data. Attention value smaller than 0.01 is omitted. Different color indicates different elements: black: C, blue: N, red: O, green: Cl, yellow: S, sky-blue: F. First row: the molecules with trifluoromethyl. Second row: the molecules with cyanide

Table 2. Ablation study on the variants of CD-MVGNN

	ToxCast	SIDER	ClinTox
No all	0.718	0.644	0.852
Only attention	0.728	0.646	0.901
Only disagreement loss	0.722	0.648	0.863
MVGNN	**0.731**	**0.652**	**0.907**
CD-MVGNN	**0.744**	**0.657**	**0.923**

The AUC-ROC scores of our methods are marked as bold.

Moreover, we find that both attention and disagreement loss can boost the performance compared with 'No All' method. Particularly, when the self-attention mechanism is employed, the performance has a significant boost, which proves that the molecular property is affected by the various atoms differently. Hence, the weights of atoms should not be considered equivalently. Thus, the proposed MVGNN and CD-MVGNN models that adopt both disagreement loss and self-attention outperform the other variants, indicating that the combination of them would significantly facilitate the model training.

3.4 Visualization of interpretability results

To illustrate the interpretability power of proposed models, we visualize certain molecules with the learned attention weights of CD-MVGNN associated with each atom within one molecule. Clintox dataset is used as an example for the demonstration, where the labels are toxicity.

Figure 5 instantiates the graph structures of the molecules along with the corresponding atom attentions. The attention values lower than 0.01 are omitted. We observe that different atoms indeed react distinctively: (i) most carbon (C) atoms that are responsible for constructing the molecule topology have got zero attention value. It is because these kinds of substructures usually do not affect the toxicity of a compound. (ii) Beyond that, the learned attention weights may indicate the functional groups that are related to the toxicity. Take Figure 5 as an example; for the upper two molecules, the C atom surrounded by three F atoms has received significantly high attentions, and all of them form a trifluoromethyl group, which is generally known responsible for the toxicity. Similarly, for the lower two molecules, the atoms with high attentions are within the cyanide. These high attention values can be helpful for discovering important atoms or potential functional groups.

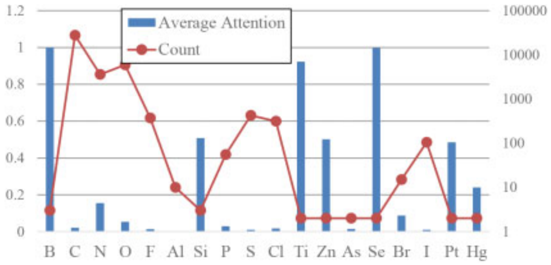

Fig. 6. Statistics of attentions in ClinTox. Left axis: the average attention value of the element. Right axis: the count of the element

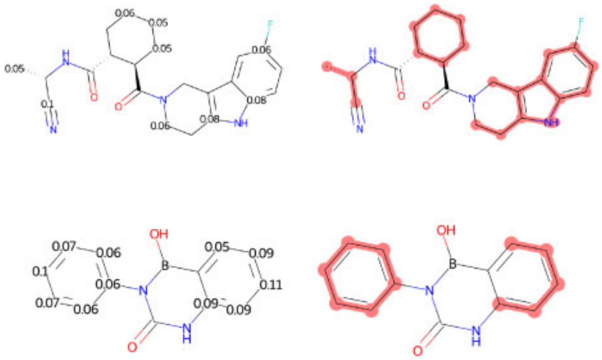

Fig. 7. Visualization of attention values versus maximally informative subgraph on Lipo data

Furthermore, we provide comprehensive statistics of the attention values over the entire ClinTox dataset. Figure 6 demonstrates the average attention values and the total occurrences of each element. It is notable that (i) atoms with high frequency do not receive high attention. For example, atom C is an essential element to maintain the molecular topology, yet it does not have significant impact on the toxicity. (ii) Atoms with low frequency but high attention values are generally heavy elements. For example, mercury (Hg) is widely known for its toxicity. The accompanied attention value of Hg is relevantly high because it usually affects the toxic property greatly.

In order to further demonstrate the effectiveness and interpretability of the self-attentive component, we also conduct a subgraph recognition experiment. A recent study (Yu *et al.*, 2020) proposes a graph information bottleneck framework to recognize the maximally informative subgraph, which can be indicated as the substructures that have the most similar property to the input molecules. We run their provided code on Lipo dataset, and visualize the recognized substructures. As shown in Figure 7, the highlights in the right two molecules demonstrate the recognized substructures, which express the most similar property, while the left two molecules with numbers illustrate the learned attention values from our model. As observed, the relatively high attention weights learned by our model are generally presented in the highlighted substructures as well. Overall, the case studies show that the proposed CD-MVGNN model is able to provide reasonable interpretability results for the prediction results, which is crucial for real drug discovery.

4 Conclusion

We propose a novel cross-dependent graph neural network (CD-MVGNN) for molecular property prediction, which is deployed via a multi-view architecture (MVGNN). Unlike previous attempts focusing exclusively on either atom-oriented graph structures or bond-oriented graph structures, our method, inspired by multi-view learning, takes both atom and bond information into consideration. Most importantly, we develop a cross-dependent message-passing scheme to allow concurrent circulation between the two views during the training in CD-MVGNN. Such approach ensures the information flow stay updated for each GNN aggregation step, which boosts the efficiency of generalized GNN, as well as increases the expressive power of CD-MVGNN. Extensive experiments against SOTA models demonstrate that proposed models outperform all baselines significantly, as well as equip with strong robustness. Although our method has achieved remarkable performance, there still exist some limitations. The two components utilized for information circulation are directly inserted without considering expert knowledge. And our method does not consider the 3D structure information. Our next step is to include expert knowledge to present a more interpretable and meaningful GNN, as well as include more information by exploring the molecule 3D structure.

Funding

This work was partially supported by US National Science Foundation IIS-1553687 and Cancer Prevention and Research Institute of Texas (CPRIT) award (RP190107).

Conflict of Interest: none declared.

References

Bhal,S.K. (2007) LogP—making sense of the value. Advanced Chemistry Development, Toronto, ON, Canada, pp. 1–4.

Bian,T. *et al.* (2020) Rumor detection on social media with bi-directional graph convolutional networks. *Proc. AAAI Conf. Artif. Intell.*, **34**, 549–556.

Chang,H. *et al.* (2021) Spectral graph attention network with fast Eigen-approximation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. Virtual Event Queensland, pp. 2905–2909.

Duvenaud,D.K. *et al.* (2015) Convolutional networks on graphs for learning molecular fingerprints. In: *NeurIPS*. Montreal, Canada, pp. 2224–2232.

Friedman,J. *et al.* (2000) Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Stat.*, **28**, 337–407.

Gilmer,J. *et al.* (2017) Neural message passing for quantum chemistry. In: *ICML*. Sydney, Australia, pp. 1263–1272.

Guo,Y. and Wu,J. (2020) Bagging MSA learning: Enhancing low-quality PSSM with deep learning for accurate protein structure property prediction. In: Proceedings of the 24th International Conference on Research in Computational Molecular Biology. Padua, Italy, Virtual Event.

Harary,F. and Norman,R.Z. (1960) Some properties of line digraphs. *Rend. Circ. Mat. Palermo*, **9**, 161–168.

Kearnes,S. *et al.* (2016) Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.*, **30**, 595–608.

Kim,S. *et al.* (2016) PubChem substance and compound databases. *Nucleic Acids Res.*, **44**, D1202–D1213.

Li,J. *et al.* (2019) Semi-supervised graph classification: a hierarchical graph perspective. In: *TheWebConf*. San Francisco, USA, pp. 972–982. ACM.

Liu,S. *et al.* (2019) N-gram graph: simple unsupervised representation for graphs, with applications to molecules. In: *NeurIPS*. Vancouver, Canada, pp. 8464–8476.

Lu,C. *et al.* (2019) Molecular property prediction: a multilevel quantum interactions modeling perspective. *Proc. AAAI Conf. Artif. Intell.*, **33**, 1052–1060.

Pires,D.E. *et al.* (2015) pkCSM: predicting small-molecule pharmacokinetic and toxicity properties using graph-based signatures. *J. Med. Chem.*, **58**, 4066–4072.

Preuer,K. *et al.* (2019) Interpretable deep learning in drug discovery. In: In: Samek W., Montavon G., Vedaldi A., Hansen L., Müller KR. (eds) *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, Cham, pp. 331–345.

Raju,A. *et al.* (2020) Graph attention multi-instance learning for accurate colorectal cancer staging. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Lima, Peru, Virtual Event, pp. 529–539. Springer.

Ramakrishnan,R. *et al.* (2015) Electronic spectra from TDDFT and machine learning in chemical space. *J. Chem. Phys.*, **143**, 084111.

Richard,A.M. *et al.* (2016) ToxCast chemical landscape: paving the road to 21st century toxicology. *Chem. Res. Toxicol.*, **29**, 1225–1251.

Rong,Y. *et al.* (2020) Self-supervised graph transformer on large-scale molecular data. In: *NeurIPS*. Virtual Event.

Schütt,K. *et al.* (2017) SchNet: a continuous-filter convolutional neural network for modeling quantum interactions. In: *NeurIPS*. Long Beach, USA, pp. 991–1001.

Song,Y. *et al.* (2020) Communicative representation learning on attributed molecular graphs. In: *Proceedings* of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020). Yokohama, Japan, Virtual Event, pp. 2831–2838.

Swamidass,S.J. *et al.* (2009) Influence relevance voting: an accurate and interpretable virtual high throughput screening method. *J. Chem. Inf. Model.*, **49**, 756–766.

Veličković,P. *et al.* (2018) Graph attention networks. In: *International Conference on Learning Representations*. Vancouver, Canada.

Wang,S. *et al.* (2019a) Graph convolutional nets for tool presence detection in surgical videos. In: *International Conference on Information Processing in Medical Imaging*. Springer, Hong Kong, China, pp. 467–478.

Wang,S. *et al.* (2019b) 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. Niagara Falls, USA, pp. 429–436.

Weisfeiler,B. and Lehman,A.A. (1968) A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno Techn. Inform.*, **2**, 12–16.

Wu,Z. *et al.* (2018) MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.*, **9**, 513–530.

Xiong,Z. *et al.* (2020) Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *J. Med. Chem.*, **63**, 8749–8760.

Xu,K. *et al.* (2018) How powerful are graph neural networks? In *International Conference on Learning Representations. Vancouver, Canada.*

Xu,Z. *et al.* (2017) Seq2seq fingerprint: an unsupervised deep molecular embedding for drug discovery. In: *Proceedings of the ACM BCB*. Boston, USA.

Yan,C. *et al.* (2020) RetroXpert: decompose retrosynthesis prediction like a chemist. In: NeurIPS. Virtual Event.

Yang,J. *et al.* (2021) Hierarchical graph capsule network. In: Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, Vol. 35. pp. 10603–10611.

Yang,K. *et al.* (2019) Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.*, **59**, 3370–3388.

Yu,J. *et al.* (2020) Graph information bottleneck for subgraph recognition. In: International Conference on Learning Representations. Addis Ababa, Ethiopia, Virtual Event.

Zeng,R. *et al.* (2019) Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, Korea, pp. 7094–7103.

Zhao,J. *et al.* (2017) Multi-view learning overview: recent progress and new challenges. *Inf. Fusion*, **38**, 43–54.

Zhao,K. *et al.* (2021) Finding critical users in social communities via graph convolutions. *IEEE Trans. Knowl. Data Eng.*, 1.