Multi-View Graph Neural Networks for Molecular Property Prediction

Hehuan Ma
University of Texas at Arlington
Arlington, TX 76019
hehuan.ma@mavs.uta.edu

Yatao Bian
Tencent AI Lab
Shenzhen, China 518057
yatao.bian@gmail.com

Yu Rong
Tencent AI Lab
Shenzhen, China 518057
yu.rong@hotmail.com

Wenbing Huang
Department of Computer Science and Technology
Tsinghua University
Beijing, China
hwenbing@126.com

Tingyang Xu
Tencent AI Lab
Shenzhen, China 518057
tingyangxu@tencent.com

Weiyang Xie
Tencent AI Lab
Shenzhen, China 518057
weiyangxie@tencent.com

Geyan Ye
Tencent AI Lab
Shenzhen, China 518057
blazerye@tencent.com

Junzhou Huang*
University of Texas at Arlington
Arlington, TX 76019
jzhuang@uta.edu

Abstract

The crux of molecular property prediction is to conduct meaningful and informative representations of the molecules. We propose **M**ulti-**V**iew **G**raph **N**eural **N**etworks (MVGNN), a multi-view architecture for generating accurate molecular property prediction by utilizing both atoms and bonds information simultaneously. A shared self-attentive readout and disagreement loss are designed to stabilize the training process and enhance the interactions between the multi-views. The visualization of the designed self-attentive readout component also provides the interpretability for the prediction results, which is crucial for real-world applications like molecular design and drug discovery. Extensive experiments on 11 datasets demonstrate the superiority and effectiveness of the proposed MVGNN model.

1 Introduction

To date, Graph Neural Networks (GNN) have gained more and more attention due to its capability of dealing with graph structured data. Molecular property prediction is also a promising application of GNN since a molecule could be represented as a graph structure by treating atoms as nodes, and bonds as edges. Despite the fruitful results obtained by GNN, there remains two limitations for current GNN models when applying them to molecular property prediction: 1) Most of the GNN models only focus on the embedding of nodes. It is truthful that nodes play an dominant role in many graph-based scenarios including social network [29, 11], recommendation system [17, 20], knowledge-graphs [8, 9], and so on. However, in some cases, nodes and edges play the equally important roles. Especially, molecular property prediction also demands information from both atoms

Machine Learning for Molecules Workshop at NeurIPS 2020. https://ml4molecules.github.io

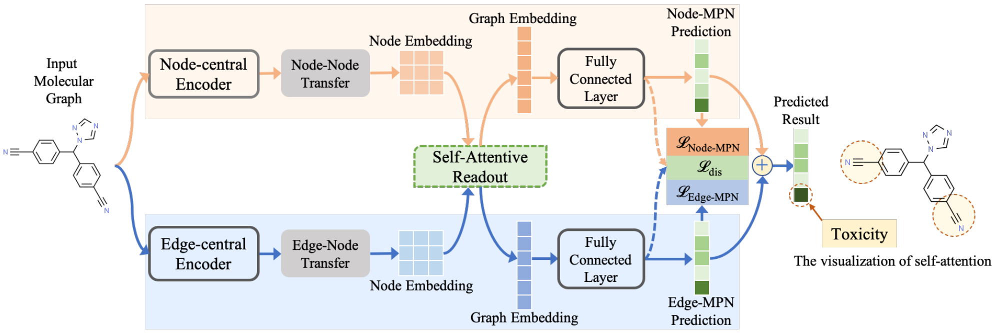

Figure 1: Overview of the proposed MVGNN architecture.

and bonds to generate precise graph embedding and make accurate prediction. Therefore, *how to properly integrate both node and edge information in a single model* is the first challenging issue. 2) the second limitation is the interpretability of the models. It is no doubt that interpretability power is very important for drug discovery. Understanding how the underlying model works will also help people figure out the key components for determining certain properties[21]. Consequently, *how to provide the explanation of the prediction results* is the second challenging issue.

To address those challenges, we believe that a fresh perspective of viewing the graph from two aspects covering both nodes and edges would be more meaningful and precise. In this paper, we take the Message Passing Neural Network (MPNN) [7] as the backbone[1], and propose a new architecture: MVGNN based on the popularity of multi-view learning, which considers the diversity of different aspects for one single target [28].

2 Preliminaries on Molecular Graph Representations

An essential preliminary is to represent a molecule to a graph representation, and extract the initial features. We view a molecule c as a graph $G_c = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = p$ refers to a set of p nodes (atoms) in the molecule and $|\mathcal{E}| = q$ refers to a set of q edges (bonds) in the molecule satisfying $(v_i, v_j) \in \mathcal{E}$. $\mathcal{N}_v$ represents the neighborhood set of node v in the graph. We denote the features of node v as $\boldsymbol{x}_v \in \mathbb{R}^{d_n}$ and the features of edge (v, k) as $\boldsymbol{e}_{vk} \in \mathbb{R}^{d_e}$ [2], where d_n and d_e refer to the feature dimension of nodes and edges, respectively. One possible node feature and edge feature are the initial chemical relevant features such as atomic mass and bond type. Please refer to Appendix A for more detailed feature extraction process. We denote a property $\boldsymbol{y}$ as the target of the predictive task, which are either binary values for classification tasks or real values for regression tasks depends on the property type. Therefore, the molecular property prediction problem can be formulated as:

Definition 2.1. Given a molecule c and its graph G_c, molecular property prediction aims to predict the property $\boldsymbol{y}_c$ according to the graph representation ξ_c that is mapped from G_c.

3 Multi-View Graph Neural Network (MVGNN)

MVGNN considers atom features and bond features equally important for constituting a molecular representation vector based on its graph structure. As demonstrated in Figure 1, MVGNN architecture contains two concurrent phases, **Node-central Encoder** and **Edge-central Encoder**, which are responsible for generating node/edge embedding matrix from the graph topology and the node/edge features. Here, we employ the message passing neural network[7], which has achieved remarkable success in modeling molecules, as the backbone to design **Node-central Encoder** and **Edge-central Encoder**, respectively. Next, MVGNN adopts **self-attentive aggregation** to learn the different weights of each embedding to produce the graph embedding. Furthermore, we share the self-attentive aggregation layer between Node-central Encoder and Edge-central Encoder to reinforce the learning of node information and edge information, respectively. After the self-attentive aggregation, MVGNN feeds the graph embeddings from Node-central Encoder and Edge-central Encoder to two independent fully connected (FC) layers to fit a loss function depending on the concrete prediction task,

[1] Here, the backbone can be any valid GNNs depending on the applications

[2] Without ambiguous, $\boldsymbol{e}_{vk}$ can represent either the edge or the edge features.

respectively. To stabilize the training process of this dual architecture, we employ a **Disagreement Loss** to enforce the outputs of two fully connected layer similar with each other.

3.1 Node-central and Edge-central Encoders

The message passing neural network (MPNN) is originally proposed in [7], which can be viewed as the simulation of information diffusion in graphs. Specially, it aggregates and passes the feature information of corresponding neighbor nodes to produce the new embeddings. Taking MPNN as the backbone, we define the Node-central and Edge-central Encoders, respectively. Specifically, the Node-MPN in the node-central encoder is defined as follows:

$$\boldsymbol{m}_v^{(l+1)} = \sum_{u\in\mathcal{N}_v} \mathsf{CONCAT}(\boldsymbol{h}_v^{(l)}, \boldsymbol{h}_u^{(l)}, \boldsymbol{e}_{vu}), \quad \boldsymbol{h}_v^{(l+1)} = \sigma(\boldsymbol{W}_{\text{node}}\boldsymbol{m}_v^{(l+1)} + \boldsymbol{h}_v^{(0)}), \tag{1}$$

where $\boldsymbol{h}_v^{(0)} = \sigma(\boldsymbol{W}_{\text{nin}}\boldsymbol{x}_v)$ is the input state of Node-MPN, and $\boldsymbol{W}_{\text{nin}} \in \mathbb{R}^{d_{\text{hid}}\times d_n}$ is the input weight matrix with an input dimension, d_{hid}. The messaging passing process in Node-MPN contains L steps. At $l+1$ step, Node-MPN updates the state of node v by aggregating the previous state of its neighbor node $u \in \mathcal{N}_n$ as well as itself, and takes the corresponding edge features $\boldsymbol{e}_{vu}$ as attached features to generate the new state of node v. Here $\boldsymbol{W}_{\text{node}} \in \mathbb{R}^{d_{\text{hid}}\times(d_e+d_{\text{hid}})}$ is the weight matrix shared in all steps and $\sigma(\cdot)$ is the activation function[3].

After L step message passing, we utilize an additional message passing step (Node-Node Transfer in Figure 1) with different weight matrix $\boldsymbol{W}_{\text{nout}} \in \mathbb{R}^{d_{\text{out}}\times(d_e+d_{\text{hid}})}$ to produce the final node embedding:

$$\boldsymbol{m}_v^{\text{o}} = \sum_{k\in\mathcal{N}_v} \mathsf{CONCAT}(h_k^{(L)}, x_k) \quad \boldsymbol{h}_v^{\text{o}} = \sigma(\boldsymbol{W}_{\text{nout}}\boldsymbol{m}_v^{\text{o}}). \tag{2}$$

By adding this additional message passing process, we can introduce more parameters as well as the non-linearity to enhance the description power of Node-MPN. We denote $\boldsymbol{H}_n = [\boldsymbol{h}_1^{\text{o}}, \cdots, \boldsymbol{h}_p^{\text{o}}] \in \mathbb{R}^{d_{\text{out}}\times p}$ as the output embeddings of Node-MPN, where d_{out} is the dimension of output embedding.

Refer to the line graph $L(G)$ of graph G in graph theory[10], the nodes can be viewed as the connections while edges can be viewed as entities. Therefore, it's possible to performing the message passing through edges to imitate Node-MPN on $L(G)$. Namely, given a edge (v, w), The Edge-based MPNN (Edge-MPN) in the edge-central encoder is formulated as:

$$\boldsymbol{m}_{vw}^{(l+1)} = \sum_{u\in\mathcal{N}_v\setminus w} \mathsf{CONCAT}(\boldsymbol{h}_{vw}^{(l)}, \boldsymbol{h}_{uv}^{(l)}, \boldsymbol{x}_u) \quad \boldsymbol{h}_{vw}^{(l+1)} = \sigma(\boldsymbol{W}_{\text{edge}}\boldsymbol{m}_{vw}^{(l+1)} + \boldsymbol{h}_{vw}^{(0)}), \tag{3}$$

where $\boldsymbol{h}_v^{(0)} = \sigma(\boldsymbol{W}_{\text{ein}}\boldsymbol{e}_{vw})$ is the input state of Edge-MPN. $\boldsymbol{W}_{\text{ein}} \in \mathbb{R}^{d_{\text{hid}}\times d_e}$ is the input weight matrix. In (3), the state vector is defined on edge e_{vw}, and the neighbor edge set of e_{vw} is defined by all edges connected to the start node v except node w. The attached features are the node features $\boldsymbol{x}_k$. The message passing and state update phase is similar with Node-MPN. Edge-MPN also contains L steps, and one more round message passing on nodes is employed to transform edge-wise embedding to node-wise (Edge-Node Transfer in Figure 1), and generate another node embedding:

$$\boldsymbol{m}_v^{\text{o}} = \sum_{k\in\mathcal{N}_v} \mathsf{CONCAT}(\boldsymbol{h}_{kv}^{(L)}, \boldsymbol{x}_k) \quad \boldsymbol{h}_v^{\text{o}} = \sigma(\boldsymbol{W}_{\text{eout}}\boldsymbol{m}_v^{\text{o}}), \tag{4}$$

where $\boldsymbol{W}_{\text{eout}} \in \mathbb{R}^{d_{\text{out}}\times(d_n+d_{\text{hid}})}$ is the weight matrix. Therefore, the final output of Edge-MPN is represented as $\boldsymbol{H}_e = [\boldsymbol{h}_1^{\text{o}}, \cdots, \boldsymbol{h}_p^{\text{o}}] \in \mathbb{R}^{d_{\text{out}}\times p}$.

3.2 The Self-attentive Readout for Graph-level Embedding

To obtain the graph representation with fixed length, a readout transformation is needed to eliminate the obstacle of node size variance and permutation variance. Here, we employ the self-attention mechanism, which is introduced in [29, 15], to learn the node importance as well as encode node embedding into a size-invariant graph embedding vector. Namely, given the output of Node-central Encoder $\boldsymbol{H}_n \in \mathbb{R}^{d_{out}\times p}$, the self-attention readout over nodes is defined as:

$$\boldsymbol{S} = \text{softmax}\left(\boldsymbol{W}_2 \tanh\left(\boldsymbol{W}_1\mathbf{H}_n\right)\right), \quad \boldsymbol{\xi}_n = \mathsf{Flatten}(\boldsymbol{S}\boldsymbol{H}_n^\top), \tag{5}$$

[3] Without any specification, we use ReLU as the activation function by default.

where $\in \mathbb{R}^{d_{\text{attn}}\times d_{\text{out}}}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{r\times d_{\text{attn}}}$ are learnable matrices, which are shared between two sub-models to enable the feature information binding and communicating during the multi-view training process. Based on the obtained self-attention $\boldsymbol{S}$, the **size invariant** and **node importance involved** graph embedding $\boldsymbol{\xi}$ is generated. Furthermore, this self-attentive readout can bring the interpretability of MVGNN as it indicates the contributions of the nodes for the downstream tasks.

3.3 The Loss of MVGNN

We feed the graph embeddings obtained from Node-central and Edge-central Encoders $\boldsymbol{\xi}_n$ and $\boldsymbol{\xi}_e$ into two distinct fully connected neural networks to obtain the predictions of two encoders respectively. We bring in a disagreement loss to minimize the difference between the two predictions, since the graph embeddings from two encoders can be viewed as the different aspects for one single target (the molecule)[28]. Therefore, no matter how the graph embedding is generated, the predictions of this single target ought to be the same. Formally, given the molecular graph set $\mathcal{G} = \{G_i\}_{i=1}^{K}$ and corresponding labels $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i=1}^{K}$, we formulate this molecular property prediction loss as follows:

$$\mathcal{L}_{\mathsf{MVGNN}} = \mathcal{L}_{\text{pred}} + \lambda\mathcal{L}_{\text{dis}}, \quad \mathcal{L}_{\text{dis}} = \sum_{G_i\in\mathcal{G}} \left|\gamma_{n,i} - \gamma_{e,i}\right|^2, \tag{6}$$

where $\mathcal{L}_{\text{pred}}$ is the supervised loss which can be formulated as $\sum_{G_i\in\mathcal{G}}(\mathcal{L}_{\mathsf{Node\text{-}GNN}}(\mathbf{y}_i, \boldsymbol{\gamma}_{n,i}) + \mathcal{L}_{\mathsf{Edge\text{-}GNN}}(\mathbf{y}_i, \boldsymbol{\gamma}_{e,i}))$, $\gamma_{*,i}$ is the output prediction produced by a fully connected neural network given graph embedding $\boldsymbol{\xi}_{*,i}$, i.e., $\gamma_{*,i} = \mathsf{ffn}(\xi_{*,i})$, $* =$ n,e. $\mathcal{L}_{\text{dis}}$ is the disagreement loss for the two predictions. λ is a hyper-parameter indicating the coefficient of the disagreement loss.

4 Experiments & Results

We compare proposed model with 7 baselines over 11 benchmark datsets. 6/11 are classification tasks, and the results are shown in Table 1. The rest are regression tasks, which are shown in Table 5 of Appendix B.3. All classification tasks are evaluated by AUC-ROC. For the regression task, we apply MAE and RMSE according to different datasets. Noted that we apply the **scaffold splitting** for all tasks on all datasets. **Scaffold splitting** splits the molecules with distinct two-dimensional structural frameworks into different subsets[1], which is more meaningful and consequential for molecular property prediction. To alleviate the effects of randomness and over-fitting, as well as to boost the robustness of the experiments, we apply cross-validation on all the experiments. All of our experiments run 10 randomly-seeded 8:1:1 data splits, which follows the same protocols in [31]. Details of the datasets, baseline models, and evaluation metric are shown in Appendix B.

Table 1: The performance comparison of classification tasks on AUC-ROC (higher is better).

Method	BACE	BBBP	Tox21	ToxCast	SIDER	ClinTox
TF_Robust [23]	$0.824_{\pm 0.022}$	$0.860_{\pm 0.087}$	$0.698_{\pm 0.012}$	$0.585_{\pm 0.031}$	$0.607_{\pm 0.033}$	$0.765_{\pm 0.085}$
GraphConv [4]	$0.854_{\pm 0.011}$	$0.877_{\pm 0.036}$	$0.772_{\pm 0.041}$	$0.650_{\pm 0.025}$	$0.593_{\pm 0.035}$	$0.845_{\pm 0.051}$
Weave [12]	$0.791_{\pm 0.008}$	$0.837_{\pm 0.065}$	$0.741_{\pm 0.044}$	$0.678_{\pm 0.024}$	$0.543_{\pm 0.034}$	$0.823_{\pm 0.023}$
SchNet [26]	$0.750_{\pm 0.033}$	$0.847_{\pm 0.024}$	$0.767_{\pm 0.025}$	$0.679_{\pm 0.021}$	$0.545_{\pm 0.038}$	$0.717_{\pm 0.042}$
MGCN [16]	$0.734_{\pm 0.030}$	$0.850_{\pm 0.064}$	$0.707_{\pm 0.016}$	$0.663_{\pm 0.009}$	$0.552_{\pm 0.018}$	$0.634_{\pm 0.042}$
Node-MPN [7]	$0.815_{\pm 0.044}$	$0.913_{\pm 0.041}$	$0.808_{\pm 0.024}$	$0.691_{\pm 0.013}$	$0.595_{\pm 0.030}$	$0.879_{\pm 0.054}$
Edge-MPN [31]	$0.852_{\pm 0.053}$	$0.919_{\pm 0.030}$	$0.826_{\pm 0.023}$	$0.718_{\pm 0.011}$	$0.632_{\pm 0.023}$	$0.897_{\pm 0.040}$
MVGNN	$\mathbf{0.863}_{\pm 0.002}$	$\mathbf{0.938}_{\pm 0.003}$	$\mathbf{0.833}_{\pm 0.001}$	$\mathbf{0.729}_{\pm 0.006}$	$\mathbf{0.644}_{\pm 0.003}$	$\mathbf{0.930}_{\pm 0.003}$

Performance Evaluation As shown in Table 1, we have these findings: (1) Clearly, our MVGNN gains significant enhancement against SOTAs on all datasets consistently with a 1.85% average improvement, which is regarded as a remarkable boost considering the challenges on these benchmarks. (2) Compared with Node-MPN and Edge-MPN, MVGNN has obtained better prediction performance as well as equipped with much smaller standard deviation. It indicates that MVGNN is not only able to generate more informative and meaningful molecular representation by considering nodes and edges simultaneously, but also more robust than both Node-MPN and Edge-MPN.

Visualization of the Interpretability Results To illustrate the interpretability of MVGNN, we visualize certain molecules with the learned attentions associated with each atom within one molecule from ClinTox dataset with toxicity as the label. As shown in Figure 2, we observe that different

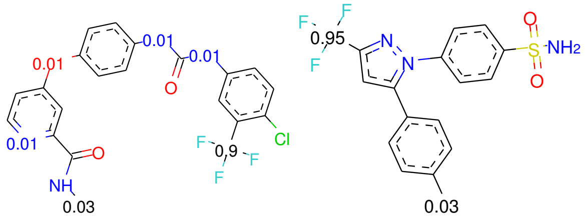

Figure 2: Visualization of attention values on selected molecules with trifluoromethyl in `ClinTox` dataset. Attention value smaller than 0.01 is omitted. Different color indicates different elements: black: C, blue: N, red: O, green: Cl, yellow: S, sky-blue: F.

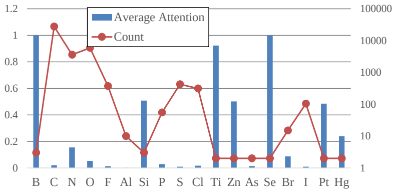

Figure 3: The statistics of attention in `ClinTox`. Left axis: the average attention value of the element. Right axis: the count of the element.

atoms indeed react distinctively: **1).** Most atom carbon (C) that are responsible for constructing the molecular topology have got zero attention value. **2).** Beyond that, MVGNN promotes the learning of functional group with impression on molecular toxicity, e,g,. toxic functional group **trifluoromethyl** is known for the toxicity [25], which reveals extremely high attention value in Figure 2. Furthermore, we provide a comprehensive statistics of the attention values over the entire `ClinTox` dataset. Figure 3 demonstrates the average attention values for each single element, as well as its total occurrences. It is notable that, **1).** atoms with high frequency do not receive high attention, such as topological element C. **2).** atoms with low frequency but high attention values are generally heavy elements. For example, Hg (Mercury) is widely known by its toxicity. Both above observations yield our assumption that regarding atoms should be considered with different weights.

Overall, compared with the previous models, MVGNN is able to achieve more accurate prediction performance, as well as provide strong interpretability for the prediction results, which are crucial for the real drug discovery applications.

References

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

[2] L. C. Blum and J.-L. Reymond. 970 million druglike small molecules for virtual screening in the chemical universe database GDB-13. *J. Am. Chem. Soc.*, 131:8732, 2009.

[3] John S Delaney. Esol: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences*, 44(3):1000–1005, 2004.

[4] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *NeurIPS*, pages 2224–2232, 2015.

[5] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2011.

[6] Kaitlyn M Gayvert, Neel S Madhukar, and Olivier Elemento. A data-driven approach to predicting successes and failures of clinical trials. *Cell chemical biology*, 23(10):1294–1301, 2016.

[7] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272. JMLR. org, 2017.

[8] Kelvin Guu, John Miller, and Percy Liang. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*, 2015.

[9] Will Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. In *Advances in Neural Information Processing Systems*, pages 2026–2037, 2018.

[10] Frank Harary and Robert Z. Norman. Some properties of line digraphs. *Rendiconti del Circolo Matematico di Palermo*, 9(2):161–168, May 1960.

[11] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. In *NeurIPS*, pages 4558–4567. 2018.

[12] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[13] Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079, 2015.

[14] Greg Landrum et al. Rdkit: Open-source cheminformatics, 2006.

[15] Jia Li, Yu Rong, Hong Cheng, Helen Meng, Wenbing Huang, and Junzhou Huang. Semi-supervised graph classification: A hierarchical graph perspective. In *The World Wide Web Conference*, pages 972–982. ACM, 2019.

[16] Chengqiang Lu, Qi Liu, Chao Wang, Zhenya Huang, Peize Lin, and Lixin He. Molecular property prediction: A multilevel quantum interactions modeling perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1052–1060, 2019.

[17] Mingsong Mao, Jie Lu, Guangquan Zhang, and Jinlong Zhang. Multirelational social recommendations via multigraph ranking. *IEEE transactions on cybernetics*, 47(12):4049–4061, 2016.

[18] Ines Filipa Martins, Ana L Teixeira, Luis Pinheiro, and Andre O Falcao. A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52(6):1686–1697, 2012.

[19] David L Mobley and J Peter Guthrie. Freesolv: a database of experimental and calculated hydration free energies, with input files. *Journal of computer-aided molecular design*, 28(7):711–720, 2014.

[20] Federico Monti, Michael Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *Advances in Neural Information Processing Systems*, pages 3697–3707, 2017.

[21] Kristina Preuer, Günter Klambauer, Friedrich Rippmann, Sepp Hochreiter, and Thomas Unterthiner. Interpretable deep learning in drug discovery. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 331–345. Springer, 2019.

[22] Raghunathan Ramakrishnan, Mia Hartmann, Enrico Tapavicza, and O Anatole Von Lilienfeld. Electronic spectra from tddft and machine learning in chemical space. *The Journal of chemical physics*, 143(8):084111, 2015.

[23] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

[24] Ann M Richard, Richard S Judson, Keith A Houck, Christopher M Grulke, Patra Volarath, Inthirany Thillainadarajah, Chihae Yang, James Rathman, Matthew T Martin, John F Wambaugh, et al. Toxcast chemical landscape: paving the road to 21st century toxicology. *Chemical research in toxicology*, 29(8):1225–1251, 2016.

[25] J Saarikoski and M Viluksela. Influence of ph on the toxicity of substituted phenols to fish. *Archives of environmental contamination and toxicology*, 10(6):747–753, 1981.

[26] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, pages 991–1001, 2017.

[27] Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling*, 56(10):1936–1949, 2016.

[28] Shiliang Sun. A survey of multi-view machine learning. *Neural computing and applications*, 23(7-8):2031–2038, 2013.

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

[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] 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.

A The Node/Edge Feature Extraction of the Molecules

The node/edge feature extraction contains two parts: 1) **node/edge messages**, which are constructed by aggregating neighboring nodes/edges features iteratively; 2) **molecule-level features**, which are the additional molecule-level features generated by RDKit to capture the global molecular information. It consists of 200 features for each molecule [14]. Since we focus on the model architecture part, we follow the exact same protocol of [31] for the initial node (atom) and edge (bond) features selection, as well as the 200 RDKit features generation procedure. The atom features description and size are listed in Table 2, and the bond features are documented in Table 3. The RDKit features are concatenated with the node/edge embedding, to go through the final FCs to make the predictions.

Table 2: Atom features [31].

features	size	description
atom type	100	type of atom (e.g., C, N, O), by atomic number
formal charge	5	integer electronic charge assigned to atom
number of bonds	6	number of bonds the atom is involved in
chirality	4	Unspecified, tetrahedral CW/CCW, or other.
number of H	5	number of bonded hydrogen atoms
atomic mass	1	mass of the atom, divided by 100
aromaticity	1	whether this atom is part of an aromatic system
hybridization	5	sp, sp2, sp3, sp3d, or sp3d2

Table 3: Bond features [31].

features	size	description
bond type	4	single, double, triple, or aromatic
stereo	6	none, any, E/Z or cis/trans
in ring	1	whether the bond is part of a ring
conjugated	1	whether the bond is conjugated

B Experimental Setup and Additional Results

B.1 Description of Dataset

Table 4 summaries the dataset statistics [30], including the property category, number of tasks and evaluation metrics of all datasets. Six datasets are used for classification, and five datasets for regression.

Molecular Classification Datasets. BACE dataset is collected for recording compounds which could act as the inhibitors of human β-secretase 1 (BACE-1) in the past few years [27]. The Blood-brain barrier penetration (BBBP) dataset contains the records of whether a compound carries the permeability property of penetrating the blood-brain barrier [18]. Tox21 and ToxCast [24] datasets include multiple toxicity labels over thousands of compounds by running high-throughput screening test on thousands of chemicals . SIDER documents marketed drug along with its adverse drug reactions, also known as the Side Effect Resource [13]. ClinTox dataset compares drugs approved through FDA and drugs eliminated due to the toxicity during clinical trials [6].

Molecular Regression Datasets. QM7 dataset is a subset of GDB-13, which records the computed atomization energies of stable and synthetically accessible organic molecules, such as HOMO/LUMO, atomization energy, etc. It contains various molecular structures such as triple bonds, cycles, amide, epoxy, etc [2]. QM8 dataset contains computer-generated quantum mechanical properties, e.g., electronic spectra and excited state energy of small molecules [22]. ESOL documents the solubility of compounds [3]. Lipophilicity dataset is selected from ChEMBL database, which is an important property that affects the molecular membrane permeability and solubility. The data is

Table 4: Datasets statstics.

Category	Dataset	Task	# Tasks	# Graphs/Molecules	Metric
Biophysics	BACE	Classification	1	1513	AUC-ROC
Physiology	BBBP	Classification	1	2039	AUC-ROC
	Tox21	Classification	12	7831	AUC-ROC
	ToxCast	Classification	617	8576	AUC-ROC
	SIDER	Classification	27	1427	AUC-ROC
	ClinTox	Classification	2	1478	AUC-ROC
Quantum Mechanics	QM7	Regression	1	6830	MAE
	QM8	Regression	12	21786	MAE
Physical Chemistry	ESOL	Regression	1	1128	RMSE
	Lipophilicity	Regression	1	4200	RMSE
	FreeSolv	Regression	1	642	RMSE

obtained via octanol/water distribution coefficient experiments [5]. FreeSolv dataset is selected from the Free Solvation Database, which contains the hydration free energy of small molecules in water from both experiments and alchemical free energy calculations [19].

B.2 Baselines

We thoroughly evaluate the performance of our methods with several popular baselines from both machine learning and chemistry communities. TF_Roubust [23] is a multitask model based on Deep Neural Network. GraphConv [4] is the vanilla graph convolutional model implementation by updating the atom features with its neighbor atoms features. Compared with GraphConv, Weave [12] model updates the atom features by constructing atom-pair with all other atoms, then combining the atom-pair features. SchNet [26] and MGCN [16] explore the molecular structure by utilizing the physical information, the 3D coordinates of each atom. Node-MPN [7] and Edge-MPN [31] perform the message passing scheme on atoms and bonds, respectively.

B.3 Results of Regression Tasks

As shown in Table 5, MVGNN achieves the best performance on all the regression benchmark datasets with a 11.3% average improvement. Specifically, our method relatively improves 17.9% over other models on ESOL dataset, yet again, reveals the superiority and robustness of MVGNN.

Table 5: Performance comparison on regression tasks based on **scaffold split** (smaller is better). Best score is marked as **bold**. Green cells indicate the results of proposed model.

Method	QM7	QM8	ESOL	Lipo	FreeSolv
TF_Robust [23]	120.600$_{\pm 9.600}$	0.024$_{\pm 0.001}$	1.722$_{\pm 0.038}$	0.909$_{\pm 0.060}$	4.122$_{\pm 0.085}$
GraphConv [4]	118.875 $_{\pm 20.219}$	0.021 $_{\pm 0.001}$	1.068 $_{\pm 0.050}$	0.712 $_{\pm 0.049}$	2.900 $_{\pm 0.135}$
Weave [12]	94.688 $_{\pm 2.705}$	0.022 $_{\pm 0.001}$	1.158 $_{\pm 0.055}$	0.813 $_{\pm 0.042}$	2.398 $_{\pm 0.250}$
SchNet [26]	74.204$_{\pm 4.983}$	0.020$_{\pm 0.002}$	1.045$_{\pm 0.064}$	0.909$_{\pm 0.098}$	3.215$_{\pm 0.755}$
MGCN [16]	77.623$_{\pm 4.734}$	0.022$_{\pm 0.002}$	1.266$_{\pm 0.147}$	1.113$_{\pm 0.041}$	3.349$_{\pm 0.097}$
Node-MPN [7]	112.960 $_{\pm 17.211}$	0.015 $_{\pm 0.002}$	1.167 $_{\pm 0.430}$	0.672 $_{\pm 0.051}$	2.185 $_{\pm 0.952}$
Edge-MPN [31]	105.775 $_{\pm 13.202}$	0.0143 $_{\pm 0.0023}$	0.980 $_{\pm 0.258}$	0.653 $_{\pm 0.046}$	2.177 $_{\pm 0.914}$
MVGNN	**71.325** $_{\pm 2.843}$	**0.0127** $_{\pm 0.0005}$	**0.8049** $_{\pm 0.036}$	**0.599** $_{\pm 0.016}$	**1.840** $_{\pm 0.194}$