Gradient-Norm Based Attentive Loss for Molecular Property Prediction

1st Hehuan Ma
University of Texas at Arlington
Arlington, United States
hehuan.ma@mavs.uta.edu

2nd Yu Rong
Tencent AI Lab
Shenzhen, China
yu.rong@hotmail.com

3th Boyang Liu
Carroll Senior High School
Southlake, United States
bliu709@southlakecarroll.edu

4rd Yuzhi Guo
University of Texas at Arlington
Arlington, United States
yuzhi.guo@mavs.uta.edu

5th Chaochao Yan
University of Texas at Arlington
Arlington, United States
chaochao.yan@mavs.uta.edu

6th Junzhou Huang
University of Texas at Arlington
Arlington, United States
jzhuang@uta.edu

2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-0126-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIBM52615.2021.9669671

***Abstract*—Molecular property prediction is one fundamental yet challenging task for drug discovery. Many studies have addressed this problem by designing deep learning algorithms, e.g., sequence-based models and graph-based models. However, the underlying data distribution is rarely explored. We discover that there exist easy samples and hard samples in the molecule datasets, and the overall distribution is usually imbalanced. Current research mainly treats them equally during the model training, while we believe that they shall not share the same weights since neural networks training is dominated by the majority class. Therefore, we propose to utilize a self-attention mechanism to generate a learnable weight for each data sample according to the associated gradient norm. The learned attention value is then embedded into the prediction models to construct an attentive loss for the network updating and back-propagation. It is empirically demonstrated that our proposed method can consistently boost the prediction performance for both classification and regression tasks.**

***Index Terms*—molecular property prediction, self-attention, loss function, learnable weight, bioinformatics**

I. Introduction

Molecular property prediction is crucial to drug discovery since it helps determine the functions of new drugs. To date, machine learning techniques, especially deep learning methods, have been widely and successfully used in many fields, e.g., computer vision (CV) [1]–[5], natural language processing (NLP) [6]–[8], and bioinformatics [9]–[12]. It is natural to apply deep learning on molecular property prediction too. Many studies have attempted to address molecular property prediction problem by utilizing and designing deep learning algorithms [13]–[15]. A molecule can be represented as either a sequence string (SMILES representation), or a graph structure. Therefore, sequence-based models used in NLP can be employed to predict the molecular property [16], [17], while graph-based models can be utilized on the molecular graph structure [18]–[21].

The molecules naturally contain some characteristics according to their biological structures, which leads to an inconsistency during the training of deep learning models. In specific, the properties of some molecules are easier to predict since they may have simpler structures or typical functional groups, while some molecules are difficult to predict since they may contain identical sub-structures but express opposite properties. The hard cases usually result in a bad prediction performance. Most of the commonly used datasets for molecular property prediction generally include more easy samples than hard samples, which brings in an imbalance problem for model training. Neural networks cannot harmonize such easy and hard cases perfectly since most of them adopt a batch learning method, which will be dominant by the majority class. In order to pursue better overall performance, neural networks are trained to minimize the comprehensive loss, which may leads to a result that easy samples are learned better but hard samples are barely learned. Besides, the gradient updates for those easy samples are quite little which cannot contribute much to the model training.

Similar problems have been studied in the image detection area since the target objects are usually difficult to detect due to the majority of background contexts. Several researches have addressed such problems by designing algorithms to balance the loss between easy samples and hard samples [22]–[25]. However, the molecular property prediction problem is more complicated. Unlike image detection where the target objects can be easily observed by human beings, hard samples in molecule data are impractical to recognize. Moreover, it is possible that a molecule is an easy sample for some properties but a hard sample for other properties, such a scenario may varies for different models too. Therefore, we propose an attentive loss function to enable the neural network to learn a weight for each sample according to the prediction difficulty level.

The intuition of our method is that easy samples and hard samples should be treated differently. Rather than simply assigning larger weights for hard samples and smaller weights for easy samples, a feasible algorithm should be designed. The reason is that it is not always good to enforce the network to learn hard samples, sometimes those may be outliers or extremely complicated molecules. Thus, we first calculate the relative gradient norm for each data sample to represent the prediction difficulty level, then employ a self-attention mechanism to allow the network to learn a proper weight for each sample. The learned attention values are then embedded with the prediction loss for model updating and back-propagation. The contribution of the proposed method can be summarized as: 1) to the best of our knowledge, we are the first to propose a learnable weighted loss to tackle the easy-hard sample imbalance problem; 2) extensive experiments on molecular property prediction tasks demonstrate that models with proposed attentive loss promote the prediction performance consistently; 3) our method is not limited to the prediction tasks or the format of input data. It can be easily embedded into any supervised models with any input data, e.g., sequence-based protein structure prediction, image-based medical image classification.

II. Related Work

A. Molecule Encoder Models for Property Prediction

One crucial part of addressing the molecular property prediction problem is to get an accurate vector representation of the molecule. Since the molecules can be represented as SMILES sequences or graphs, the encoder models can be either sequence-based models or graph-based models. Sequence-based models spot the potentially useful information of the molecular SMILES sequence data by adequately training them using Recurrent Neural Networks (RNNs), in order to obtain the vector representation of the molecule [16], [17], [26]. Graph-based techniques are used to utilize the graph structure of a molecule, and Graph Neural Networks (GNNs) are employed to generate the molecular representation by embedding the graph features into a continuous vector [13], [19]–[21], [27], [28]. Graph Isomorphism Network (GIN) [27] and Graph Attention Network (GAT) [28] are two representative work. In this paper, we take the graph-based models as the backbone models to verify the effectiveness of our method.

B. Loss Function Regarding Class Imbalance

Several studies have attempted to harmonize the imbalanced data distribution problem. Focal loss discovers the positive-negative sample imbalance problem in the image detection area [22]. The background negative samples are much more than the positive target samples, thus the regular dense sampling method overwhelms the model training. They propose to reshape the standard cross-entropy loss to assign smaller weights for those well-classified samples. Later, [24] points out the easy-hard sample imbalance problem for detection. They summarize the disharmonies with regards to the distribution of the gradient norm and design a gradient harmonizing mechanism (GHM) to modify the gradients by reformulating the loss function.

III. Methodology

The implementation of our proposed method is introduced in this section. The overview of the entire model framework is illustrated in Fig. 1.

A. Molecular Property Prediction Model

1) Problem Definition: The molecular property prediction problem is a prediction task that includes classification and regression problem. Given a molecule $\mathcal{M}$, it contains property y, where $y \in \{0, 1\}$ for classification problem and $y \in \mathbb{R}$ for regression problem. The commonly used representation of molecule $\mathcal{M}$ refers to either a sequence or a graph. For sequence-based input, $\mathcal{M}$ is represented by SMILES, and language models are applied to convert the SMILES string to a feature vector $\mathbf{h}_s \in \mathbb{R}^{d_s}$, where d_s is the dimension of sequence-based features. For graph-based input, the graph structure of $\mathcal{M}$ is usually extracted by RDKit [29]. Then GNN-based models are employed to learn the vector representation $\mathbf{h}_g \in \mathbb{R}^{d_g}$ according to the graph features, where d_g is the dimension of graph-based features. In our experiments, we take the graph structure of $\mathcal{M}$ as the input and take GIN [27] and GAT [28] as the backbone models to conduct extensive experiments.

2) Graph-based Encoder Module: Molecule $\mathcal{M}$ can be naturally represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = p$ refers to the set of p atoms and $|\mathcal{E}| = q$ refers to a set of q bonds within the molecule. The features of atom v is referred as $\mathbf{a}_v \in \mathbb{R}^{d_a}$, and the features of bond (v, u) is referred as $\mathbf{b}_{vu} \in \mathbb{R}^{d_b}$, where $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_b}$ represent the feature dimension of atom and bond respectively. $\mathcal{N}(v)$ denotes the neighbor atoms of atom v, which is identified by the bonds between atoms.

Most of the commonly used GNN-based models follow a procedure of message passing and state update. In specific, the state of the target node v at layer/iteration l $(l = 0, 1, \ldots, L)$ is updated by aggregating the information of its neighborhood $\mathbf{h}_u$ $(u \in \mathcal{N}(v))$, then combined with the state of itself $\mathbf{h}_v$. After L layer/iteration, the states of all the nodes are captured to generate a vector representation $\mathbf{h}_\mathcal{G}$ through a readout mechanism. The process can be formulated as:

$$\mathbf{h}_{\mathcal{N}(v)}^{(l)} = \text{AGGREGATE}_{\ l} \left(\left\{ \mathbf{h}_u^{(l-1)}, \forall u \in \mathcal{N}(v) \right\} \right), \quad (1)$$

$$\mathbf{h}_v^{(l)} = \sigma \left(W^{(l)} \cdot \text{CONCAT} \left(\mathbf{h}_v^{(l-1)}, \mathbf{h}_{\mathcal{N}(v)}^{(l)} \right) \right), \quad (2)$$

$$\mathbf{h}_\mathcal{G} = \text{READOUT}(\{\mathbf{h}_v^L |\ v \in \mathcal{V}\}), \quad (3)$$

where $W^{(l)}$ is the weight matrix, and σ is the activation function. The readout mechanism can be operations like summation or mean.

In this paper, we use two commonly used variants of graph-based models, GIN and GAT, as the backbone models to confirm the effectiveness of proposed method. GIN is theoretically proved as one of the most powerful GNN models. It utilizes multi-layer perceptron (MLP) for state update, as well as employ a concatenate operation over all layers/iterations during the readout phase. The updated rule in Equation 2 and 3 can be summarized as:

$$\mathbf{h}_v^{(l)} = \text{MLP}^{(l)} \left(\left(1 + \epsilon^{(l)} \right) \cdot \mathbf{h}_v^{(l-1)} + \sum_{u \in \mathcal{N}(v)} \mathbf{h}_u^{(l-1)} \right), \quad (4)$$

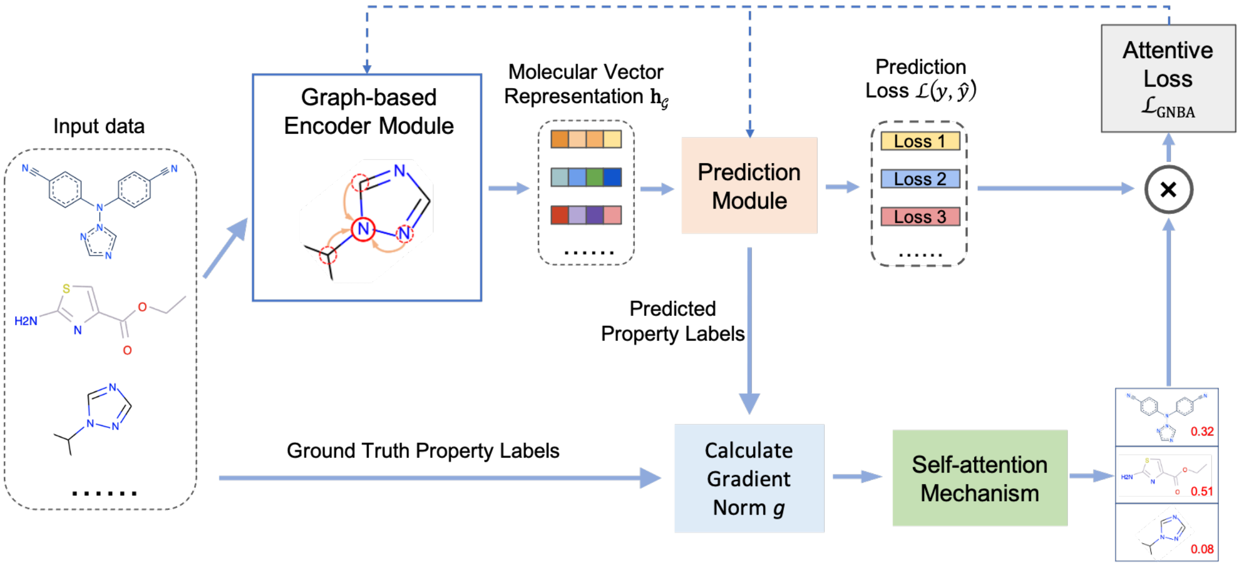

Input data
Graph-based Encoder Module
N
N
N
Molecular Vector Representation $\mathbf{h}_\mathcal{G}$
......
Prediction Module
Prediction Loss $\mathcal{L}(y, \hat{y})$
Loss 1
Loss 2
Loss 3
......
Attentive Loss $\mathcal{L}_{\text{GNBA}}$
Predicted Property Labels
Ground Truth Property Labels
Calculate Gradient Norm g
Self-attention Mechanism
0.32
0.51
0.08
H2N
S
N
O
O
......

Fig. 1. Overview of the model framework. The molecules are fed into the graph-based encoder module to generate vector representations. The vector representations are then used in the prediction module to predict the property labels. Next, the gradient norms are calculated by the predictions and the ground truth property labels, and then go through a self-attention mechanism to generate the attention values for the molecules. Last, the attentions are embedded into the prediction loss for model updating and back-propagation.

$$\mathbf{h}_\mathcal{G} = \text{CONCAT}\left(\text{READOUT}\left(\left\{\mathbf{h}_v^{(l)} \mid v \in \mathcal{V}\right\}\right)\right), \tag{5}$$

where $l = 0, 1, \ldots, L$, and ϵ is a fixed scalar or a learnable parameter.

GAT employs an attention mechanism over the neighbors of target node, thus each neighbor node gets an associated weight, e.g., more important nodes receive higher weight values. Rather than treating every neighborhood equally, GAT considers the learned attention a_{vu} along with every neighborhood during the aggregation part. Therefore, Equation 2 can be updated with

$$\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} a_{vu} W^{(l-1)} \mathbf{h}_u^{(l-1)}\right), \tag{6}$$

$$a_{vu} = \exp\left(\frac{\sigma\left(\boldsymbol{\alpha}^{\mathrm{T}}\left[W\mathbf{h}_v \| W\mathbf{h}_u\right]\right)}{\sum_{k \in N(v)} \boldsymbol{\alpha}^{\mathrm{T}}\left[W\mathbf{h}_v \| W\mathbf{h}_k\right]}\right), \tag{7}$$

where $\boldsymbol{\alpha}$ is a weight vector parameter for the attention mechanism, and $(\cdot)^T$ denotes the transposition and $\|$ represents the concatenation operation.

3) Prediction Module: After going through the graph encoder module, a graph representation $\mathbf{h}_\mathcal{G}$ is obtained and fed into the following inference module for property prediction. The prediction module can be simply one or several fully connected (FC) layers or MLP. Here we follow the same protocol used in [20], which is one FC layer:

$$\hat{y} = FC(\mathbf{h}_\mathcal{G}). \tag{8}$$

$\hat{y}$ is the output of the prediction module, which refers to the predicted probability for the classification tasks and the actual predicted property value for the regression tasks. Furthermore, we defined the supervised loss $\mathcal{L}(y, \hat{y})$ as $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ for classification and regression tasks respectively. In this paper, we used Binary Cross Entropy (BCE) loss to update the network for classification tasks:

$$\mathcal{L}_{cls} = -W\left[y \cdot \log \hat{y} + (1-y) \cdot \log(1-\hat{y})\right], \tag{9}$$

where W is the weight matrix. And Mean Squared Error (MSE) loss is applied for regression tasks:

$$\mathcal{L}_{reg} = (\hat{y} - y)^2 \tag{10}$$

B. Gradient Norm

The gradient norm g we used here is not calculated strictly following the common definition of the gradient norm during the network update. It is a relative norm of the input sample's gradient, which reflects if the sample is easy or hard to predict. The term of gradient norm g is used for convenience. Specifically, we measure the distance between the prediction and the ground truth label, and scale the value to (0,1). A similar protocol is also established in [24]. The gradient norm for classification tasks is defined as:

$$g_c = |\hat{y} - y| = \begin{cases} 1 - \hat{y} & \text{if } y = 1, \\ \hat{y} & \text{if } y = 0. \end{cases} \tag{11}$$

Since $\hat{y}$ in Equation 11 represents the predicted probability which is obtained by performing a sigmoid operation on the direct logits output from the prediction network, g_c is capable of indicating how well the sample is predicted. Moreover, we simplified the gradient norm for regression tasks as shown in Equation 12, and the value is scaled to (0,1) for better visualization.

$$g_r = (sigmoid\,|\hat{y} - y| - 0.5) \times 2. \tag{12}$$

C. Attentive Loss

The gradient-norm based attentive loss ($\mathcal{L}_{GNBA}$) is calculated based on the gradient value g. Specifically, a self-attention mechanism is applied on the calculated g during training [30], thus the attention value is learned and updated during the model training by:

$$attn = \operatorname{softmax}\left(W_2 \tanh\left(W_1 g\right)\right), \tag{13}$$

where g defers to g_c for classification and g_r for regression, $W_1 \in \mathbb{R}^{d_{\text{attn}} \times 1}$ and $W_2 \in \mathbb{R}^{1 \times d_{\text{attn}}}$ are learnable matrices, d_{attn} is the hidden dimension in the self-attention mechanism. W_1 linearly transforms the gradient norm g to a h_{attn}-dimensional space, while W_2 provides the insights of sample importance, then a softmax function is followed to normalize the importance. Thus, $\mathcal{L}_{GNBA}$ is designed by embedding the attention value $attn$ into the prediction loss for each sample. Suppose the dataset contains molecules $M = \{\mathcal{M}_i\}_{i=1}^{K}$,

$$\mathcal{L}_{GNBA} = \begin{cases} \sum_{\mathcal{M}_i \in M} \mathcal{L}_{cls} \cdot attn & \text{if } classification, \\ \sum_{\mathcal{M}_i \in M} \mathcal{L}_{reg} \cdot attn & \text{if } regression. \end{cases} \tag{14}$$

IV. Experiments and Results

In this section, we describe the implementation of extensive experiments in detail.

A. Experimental Settings

1) Implementation: Our method is implemented on top of the code from [20], which includes the construction of backbone models GIN and GAT. Our experiments are conducted in a pair-wise manner. In specific, we first conduct experiments utilizing GIN and GAT on various datasets, and then we add proposed attentive loss to each model to compare the performance difference.

2) Dataset Split: The dataset is split randomly into train/validation/test with a ratio of 8:1:1, and we ensure the data splits are exactly the same for each pair-wise experiment. All the experiments are run 3 times to alleviate the randomness as well as demonstrate the robustness. We take the average and standard deviation of the evaluation scores as the final results.

B. Dataset Description and Evaluation

We have conducted extensive experiments for both classification tasks and regression tasks. Bbbp and bace are classification datasets, while lipophilicity and esol are regression datasets.

1) Datasets: **bbbp** is the Blood-brain barrier penetration dataset [31]. The **bace** dataset is a documentation that records the compounds which may act as the inhibitors of human - secretase 1 (BACE-1) [32]. The **lipophilicity** dataset is selected from ChEMBL database, which is an important property that influences the molecular membrane permeability and solubility [33]. **Esol** stands for Estimated Solubility, it includes the aqueous solubility information of compounds [34].

2) Evaluation Metrics: In this paper, we use area under the receiver operating characteristic curve (ROC-AUC) as the evaluation criteria for all classification tasks, and root mean square error (RMSE) for all regression tasks.

3) Baselines: The experiments are conducted in a pair-wise manner to verify the effectiveness of the proposed method. Specifically, the baseline model is considered as running with GIN or GAT directly, and our method is implemented by adding the attentive loss to the baseline models. Consequently, as shown in Table I, **GIN** and **GAT** denote the baseline models, while **GIN + attn** and **GAT + attn** represent the proposed method. All the models are run on the four datasets accordingly. Moreover, we keep all the hyper-parameters exactly the same to prove the effectiveness of our method, except for the self-attention mechanism.

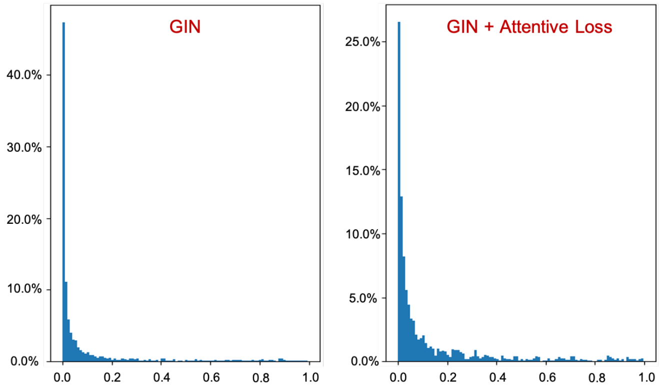


Fig. 2. An illustration of relative gradient norm distribution between GIN and GIN equipped with attentive loss on bbbp dataset. x-axis represents the values of the gradient norm, and y-axis represents the fraction of the data samples.

C. Experimental Results

1) Visualization of Gradient Norm Distribution: Based on our assumption, the gradient norm distribution of input dataset should be changed after applying proposed attentive loss. Thus, we first plot the histogram of the data according to the gradient norm after training. Fig. 2 demonstrates the distribution of the gradient norm on bbbp dataset. The figure on the left shows the distribution of employing GIN only, while the figure on the right denotes GIN equipped with the self-attention mechanism. As observed, the distribution has changed as expected. For the baseline method, there exists many easy samples with quite small gradients, thus the model training benefits little from them. In the mean time, we observed from our method that the distribution indeed trends to balance the samples by increasing the gradient for those easy samples, which promotes the model to train better.

2) Comparison Results: Extensive experiments are conducted on both classification and regression datasets using GIN and GAT as the backbone models. The results are shown in Table I in a pair-wise manner. We can see that models with proposed attentive loss outperform all the baseline models. The improvement can be up to 1.89% for the classification tasks, and 11.78% for regression tasks. Since our proposed

TABLE I

The experiment results on classification and regression datasets respectively. The better score for each method pair is marked as **bold**. Bbbp and bace are used for classification tasks, and the evaluation score is ROC-AUC, higher is better. Lipophilicity and esol are used for regression tasks, and the evaluation score is RMSE, lower is better.

	Classification		Regression	
Dataset / Method	Bbbp	Bace	Lipophilicity	Esol
GIN	0.920 ±0.011	0.898 ±0.003	0.669 ±0.010	1.074 ±0.112
GIN + attn	**0.937** ±0.002	**0.911** ±0.001	**0.591** ±0.010	**0.956** ±0.027
GAT	0.928 ±0.001	0.905 ±0.003	0.558 ±0.043	0.958 ±0.017
GAT + attn	**0.938** ±0.003	**0.912** ±0.0003	**0.527** ±0.005	**0.891** ±0.014

method can be considered as an add-on unit for any prediction model, the influence, in theory, should not be that significant. The self-attention mechanism is able to improve the prediction performance by introducing different weights for each sample according to the prediction difficulty level, while at the same time, not changing the underlying neural network algorithms.

3) Attention Values Visualization: In order to further verify our hypothesis that the network is able to learn the attentions according to the prediction difficulty and may varies from different datasets, we plot the learned attentions along with the gradient norm to check the overall trend between them. Fig. 3 demonstrates the gradient norms and the attention values for each dataset, which is generated on the training dataset from the training epoch with the best validation score, with the backbone model of GIN. Since the attention is learned per batch during model training, the value is relatively small compared with the gradient norm. We have applied max-min normalization to scale the attention value to (0,1) for better visualization. The figure is plot by sorting the gradient norms in an ascending order along with the corresponding attention values. As observed, there empirically exists certain relations between the gradient norm and the learned attention. Moreover, it varies for different datasets as expected. For dataset bace, with the increase of gradient norm, the attention is decreased; while for other datasets, both of them follow the same trend. Such observations confirm that the weights associated with each sample shall not be simply defined, it will be related to the data distribution, as well as the prediction difficulty level. For some datasets, paying more attention on the hard samples may help increase the overall prediction performance. However, for some other datasets, those extremely hard samples may be outliers so learning more from them may result in a worse performance for other samples. In consequence, a learnable weight should be adjusted for training. Our proposed method pushes the network to learn how to deal with these samples, and assign an attention value to indicate the importance of each sample. It is empirically demonstrated that our method can boost the training of any backbone models.

V. Conclusion

We propose a gradient-norm based attentive loss for molecular property prediction, which is deployed via a self-attention mechanism. Rather than developing training algorithms to improve the prediction performance, we dive into the data level to explore the relationship between each data sample, which brings in a novel perspective to address molecular property prediction in the field. Extensive experiments have confirmed the effectiveness of proposed method. Our attentive loss can also be embedded into any supervised learning models since it only depends on the relative gradient norm. Moreover, the attentive loss is fully data-driven, which means all you need is to equip it with your existing models and the network will do the rest. Our proposed method is the first step towards a data-driven weight learning mechanism to address easy-hard sample imbalance problem in molecular property prediction. Notwithstanding, there still remains unexplored perspectives in such direction, which are our future work. For example, we may perform a case-by-case analysis on the test dataset to see how the predictions change with the learnable weights, or conduct experiments on more datasets to get a comprehensive study across different molecular properties.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[4] J. Yang, P. Zhao, Y. Rong, C. Yan, C. Li, H. Ma, and J. Huang, "Hierarchical graph capsule network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 603–10 611.

[5] J. Yang, C. Li, W. An, H. Ma, Y. Guo, Y. Rong, P. Zhao, and J. Huang, "Exploring robustness of unsupervised domain adaptation in semantic segmentation," *arXiv preprint arXiv:2105.10843*, 2021.

[6] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," in *Artificial Intelligence and Statistics*, 2012, pp. 127–135.

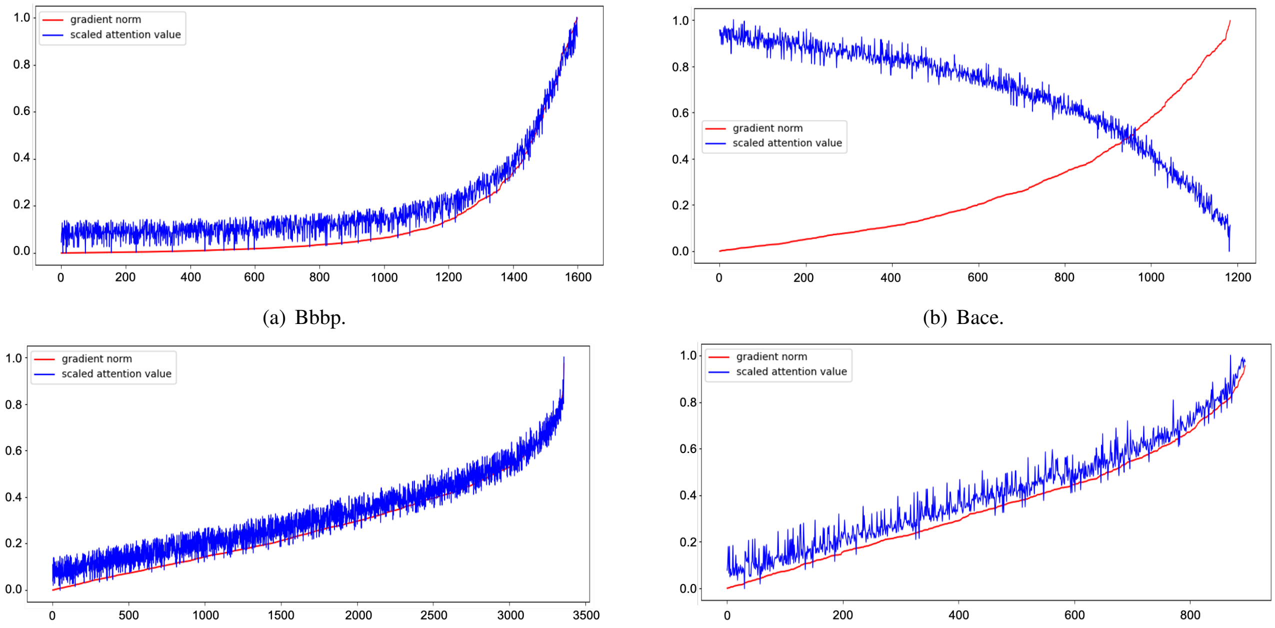
[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[8] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.

[9] H. Ma, C. Yan, Y. Guo, S. Wang, Y. Wang, H. Sun, and J. Huang, "Improving molecular property prediction on limited data with deep multi-label learning," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 2779–2784.

[10] Y. Guo, J. Wu, H. Ma, S. Wang, and J. Huang, "Bagging msa learning: Enhancing low-quality pssm with deep learning for accurate protein structure property prediction," in *International Conference on Research in Computational Molecular Biology*. Springer, 2020, pp. 88–103.

[11] H. Ma, W. An, Y. Wang, H. Sun, R. Huang, and J. Huang, "Deep graph learning with property augmentation for predicting drug-induced liver injury," *Chemical Research in Toxicology*, vol. 34, no. 2, pp. 495–506, 2020.



(a) Bbbp.

(b) Bace.

(c) Lipophilicity.

(d) Esol.

Fig. 3. Visualization of the gradient norms and the scaled attention values for each training dataset with GIN as the backbone model. x-axis denotes each training sample, and y-axis denotes the value of the sample's relative gradient norm and attention. The data is sorted in an ascending order based on the values of relative gradient norm, and the attention values are plotted accordingly.

[12] Y. Guo, J. Wu, H. Ma, J. Yang, X. Zhu, and J. Huang, "Weightaln: Weighted homologous alignment for protein structure property prediction," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 72–75.

[13] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*. JMLR. org, 2017, pp. 1263–1272.

[14] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, and J. Huang, "Self-supervised graph transformer on large-scale molecular data," *arXiv preprint arXiv:2007.02835*, 2020.

[15] Y. Guo, J. Wu, H. Ma, S. Wang, and J. Huang, "Protein ensemble learning with atrous spatial pyramid networks for secondary structure prediction," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 17–22.

[16] Z. Xu, S. Wang, F. Zhu, and J. 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*, 2017, pp. 285–294.

[17] S. Wang, Y. Guo, Y. Wang, H. Sun, and J. 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*, 2019, pp. 429–436.

[18] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[19] K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea *et al.*, "Analyzing learned molecular representations for property prediction," *Journal of chemical information and modeling*, vol. 59, no. 8, pp. 3370–3388, 2019.

[20] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," *arXiv preprint arXiv:1905.12265*, 2019.

[21] H. Ma, Y. Bian, Y. Rong, W. Huang, T. Xu, W. Xie, G. Ye, and J. Huang, "Multi-view graph neural networks for molecular property prediction," *arXiv preprint arXiv:2005.13607*, 2020.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[23] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra r-cnn: Towards balanced learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 821–830.

[24] B. Li, Y. Liu, and X. Wang, "Gradient harmonized single-stage detector," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8577–8584.

[25] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 1567–1578.

[26] X. Zhang, S. Wang, F. Zhu, Z. Xu, Y. Wang, and J. Huang, "Seq3seq fingerprint: towards end-to-end semi-supervised deep drug discovery," in *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2018, pp. 404–413.

[27] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2018.

[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[29] G. Landrum *et al.*, "Rdkit: Open-source cheminformatics," 2006.

[30] J. Li, Y. Rong, H. Cheng, H. Meng, W. Huang, and J. Huang, "Semi-supervised graph classification: A hierarchical graph perspective," in *The World Wide Web Conference*, 2019, pp. 972–982.

[31] I. F. Martins, A. L. Teixeira, L. Pinheiro, and A. O. Falcao, "A bayesian approach to in silico blood-brain barrier penetration modeling," *Journal of chemical information and modeling*, vol. 52, no. 6, pp. 1686–1697, 2012.

[32] G. Subramanian, B. Ramsundar, V. Pande, and R. A. Denny, "Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches," *Journal of chemical information and modeling*, vol. 56, no. 10, pp. 1936–1949, 2016.

[33] A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani *et al.*, "Chembl: a large-scale bioactivity database for drug discovery," *Nucleic acids research*, vol. 40, no. D1, pp. D1100–D1107, 2011.

[34] J. S. Delaney, "Esol: estimating aqueous solubility directly from molecular structure," *Journal of chemical information and computer sciences*, no. 3, pp. 1000–1005, 2004.