Communicative Representation Learning on Attributed Molecular Graphs

Ying Song[1,2†], **Shuangjia Zheng**[1†], **Zhangming Niu**[2], **Zhang-Hua Fu**[3,4], **Yutong Lu**[1] and **Yuedong Yang**[1*]
[1]Sun Yat-sen University
[2]Aladdin Healthcare Technologies Ltd
[3]The Chinese University of Hong Kong, Shenzhen
[4]Shenzhen Institute of Artificial Intelligence and Robotics for Society
{songy75, zhengshj9}@mail2.sysu.edu.cn, zhangming@aladdinid.com, fuzhanghua@cuhk.edu.cn, yutong.lu@nscc-gz.cn, yangyd25@mail.sysu.edu.cn

Abstract

Constructing proper representations of molecules lies at the core of numerous tasks such as molecular property prediction and drug design. Graph neural networks, especially message passing neural network (MPNN) and its variants, have recently made remarkable achievements in molecular graph modeling. Albeit powerful, the one-sided focuses on atom (node) or bond (edge) information of existing MPNN methods lead to the insufficient representations of the attributed molecular graphs. Herein, we propose a Communicative Message Passing Neural Network (CMPNN) to improve the molecular embedding by strengthening the message interactions between nodes and edges through a communicative kernel. In addition, the message generation process is enriched by introducing a new message booster module. Extensive experiments demonstrated that the proposed model obtained superior performances against state-of-the-art baselines on six chemical property datasets. Further visualization also showed better representation capacity of our model.

1 Introduction

Accurately predicting the properties of molecules has always been a topic of interest in the pharmaceutical community. The major goal of molecular property prediction is to remove compounds which are more likely to have property liabilities during downstream development, hence the desire to save tons of resources as well as time [Cherkasov *et al.*2014].

Briefly, the key idea of property prediction is to first map an input molecule m to a dense feature vector with a representation function, $h = g(m)$, and then make prediction of the targeted property based on the embedding by $y = f(h)$.

Early studies of quantitative structure-property relationships (QSPR) have been carried out based upon feature engineering e.g. expert-crafted physicochemical descriptors [Nettles *et al.*2007] and molecular fingerprints [Rogers and Hahn2010]. However, descriptor-based representation methods presume that all information related to the task predictions is covered in the chosen descriptor set, restricting the capability for a model to learn beyond the existing chemical knowledge.

In recent decades, with the substantial increase in available experimental molecular properties data points, machine learning especially deep learning methods have shown strong potentials to compete with or even outperform conventional approaches. Compared to the previous descriptor-based methods, deep learning-based models can take the relatively lossless 'raw' molecule formats e.g. SMILES strings and topological graphs as input, and then train models in an end-to-end fashion to predict the target endpoints. The representations obtained from these models are potentially able to profile comprehensive information for molecules.

A chemical structure could be intrinsically depicted as a hydrogen-depleted topological graphs whose nodes represent atoms with edges representing for bonds. In this sense, graph-based algorithms could be intuitively introduced to learn the representations of molecules. [Duvenaud *et al.*2015] reported a neural fingerprint method as an alternative to molecular fingerprints, and also one of the earliest efforts in employing graph convolution approaches on chemical representations. Then, several graph convolution models were reported as extensions to the Duvenaud's method by increasing molecular attributes. Recently [Gilmer *et al.*2017] summarized a general architecture called message passing neural networks (MPNNs) that demonstrated superior performance in predictions of quantum chemical properties. Broadly speaking, the MPNN framework includes three main modules: (1) message passing module, where, information of each atom is transmitted from its neighbors across the molecular graph into a message vector; (2) updating module, where the hidden states at each atom in the molecule are updated based on the obtained message vector;

However, MPNN and its variants mainly focused on obtaining effective vertices (atoms) embedding, but ignored the information carried by edges (bonds) that can be favorable to many downstream tasks such as node or edge embeddings and graph representations. To alleviate this problem, directed MPNN (D-MPNN) [Yang *et al.*2019] has been introduced to alleviate the problem by using messages associated with

[†]These two authors contributed equally.
[*]Corresponding author.

directed edges (bonds) rather than those with vertices. The main contribution of DMPNN is that it can make use of the bond attributes as well as avoid unnecessary loops in the message passing trajectory, and thus obtain information without redundancy. This bond-based message passing procedure has shown outstanding performances compared to the previous MPNNs. However, the DMPNN has ignored the passing back of messages from bonds to atoms. The failure to communicate both types of messages makes DMPNN limited to capture the complementary information of atoms and bonds efficiently.

In this study, we proposed a directed graph-based Communicative Message Passing Neural Network (CMPNN)* to improve the molecular graph embedding by strengthening the message interactions between edges (bonds) and nodes (atoms). In our framework, both the bond and atom embeddings are updated during the training process. To avoid information redundancy, we elaborately designed an node-edge interaction procedure and compared it with several variants. In addition, a message booster was introduced to enrich the message generation process. Extensive experiments demonstrated that the proposed model obtained superior performance against state-of-the-art baselines on six chemical graph datasets. Further visualization of atom embedding also showed better model representation capacity.

In summary, the main contributions of our work are as follows:

- We propose a directed graph-based Communicative Message Passing Neural Network, an efficient graph model to update the edge and node embeddings interactively.
- A message booster is introduced to enrich the message generation process, which can be generalized into other graph-related tasks such as the node classification and link prediction.
- Extensive experiments are conducted on different levels of public datasets to demonstrate the effectiveness of our method.

2 Related Work

Descriptor-based Representation. One of the most popular descriptors is the chemical fingerprints e.g. Extended-Connectivity Fingerprints (ECFP) [Rogers and Hahn2010]. These fingerprints encode the neighboring environments of heavy atoms in a compound into a fixed bit string with a hash function, where each bit indicates whether a certain substructure is present in the compound. A series of neural network-based methods with fingerprint inputs have been developed [Dahl *et al.*2014], which have been shown to significantly improve on current Random Forest or SVM based models. However, because of the requirement of large space to represent molecules, the resulting hash bit strings are usually highly sparse. Besides, the non-invertible character of hash functions makes it hard to interpret the relationship between properties and structures.

*https://github.com/SY575/CMPNN

$G = (V, E)$	Input graph.
u, v,...	Nodes in G.
$\mathbf{e}_{u,v}$	A link from node u to v.
$N(v)$	The set of neighbor nodes of node v.
$\mathbf{x}$	Raw feature.
$\mathbf{h}^i(v)$	The hidden representation of node v in layer i.
$\mathbf{h}^i(\mathrm{e}_{v,w})$	The hidden representation of edge $\mathbf{e}_{v,w}$ in layer i.
$\mathbf{W}$	Weight matrix.
σ	Acitive function.

Table 1: Mathematical notation list.

Linear Notation-based Representation. Another option for molecule representations is the molecular linear notation, among which the most common one is the SMILES notationweininger1989smiles. This linear representation encodes the topological information of a molecule based on common chemical bonding rules. Several attempts were made to feed SMILES into more complicated neural networks [Zheng *et al.*2019b] [Jastrzebski *et al.*2016] [Zheng *et al.*2020] [Zheng *et al.*2019a]. Nevertheless, the poor scalability of the sequential representation and the loss of spatial information limit the performances of these kinds of approaches.

Graph Structure-based Representation. Recent works started to explore the molecular graph representation. [Duvenaud *et al.*2015] first applied convolutional layers to encode molecular graphs into neural fingerprints. Following this work, several variants have made various extensions to work on property prediction tasks [Coley *et al.*2017], while most of them focused on atom-based message passing. To gain supplementary information from bonds, [Kearnes *et al.*2016] proposed to utilize attributes of both nodes (atoms) and edges (bonds), and [Gilmer *et al.*2017] generalized it into a MPNN framework. [Coley *et al.*2017] created atom-bond feature vectors by concatenating features of an atom and all neighboring bonds. In these works, atom attributes and bond attributes are treated homogeneously instead of with internal relations. Though a few more studies explored the information of the edges through network modules like the edge attention mechanism [Shang *et al.*2018] and edge memory module [Withnall *et al.*2020], there models were yet built upon the atom-based MPNN and thus suffered from the information redundancy during iterations. DMPNN [Yang *et al.*2019] was introduced as an alternative since it treated the molecular graph as an edge-oriented directed structure, avoiding the unnecessary loops in training procedure.

The proposed CMPNN follows the edge-based message passing in DMPNN and introduces the node-edge interaction module to leverage the node and edge attributes during message passing. To our best knowledge, this work is the first study to build node-edge communicative message passing in the directed graph.

3 Methods

The key idea behind CMPNN is that we strengthen the message interactions between bonds and atoms to obtain a bet-

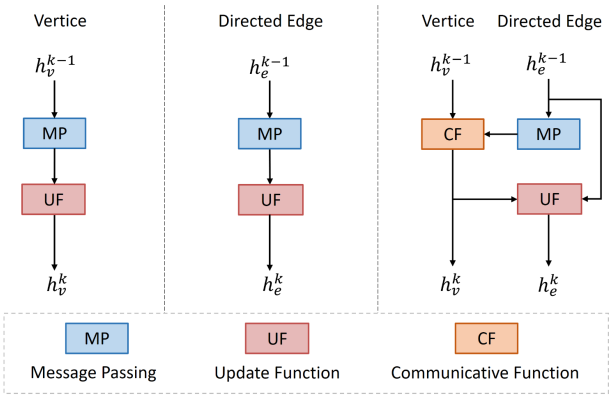


Figure 1: Comparing message passing procedure among MPNN (left), DMPNN (middle) and CMPNN (right).

ter representation of a molecular graph. We first describe the overview of communicative message passing algorithm, which updates the atom and bond messages in a directed molecular graph interactively, and aggregates all information to obtain the neural representation of a molecule (Section 3.1). We then separately introduce two augmentations, which we refer to as the message booster (Section 3.2) and node-edge message communication function (Section 3.3). Notations in this paper are listed in Table 1.

3.1 Communicative Message Passing

In this section, we describe the molecular embedding generation with communicative message passing (Algorithm 1) and compare it with original MPNN and DMPNN.

The CMPNN interactively operates on edge hidden states $\mathbf{h}(\mathrm{e}_{v,w})$, node hidden state $\mathbf{h}(v)$, message $\mathbf{m}(\mathrm{e}_{v,w})$ and $\mathbf{m}(v)$; while MPNN operates on the node $\mathbf{h}(v)$ and message $\mathbf{m}(v)$; DMPNN on the edge hidden states $\mathbf{h}(\mathrm{e}_{v,w})$ and message $\mathbf{m}(\mathrm{e}_{v,w})$. An overall comparison among three MPNNs are depicted in Figure 1.

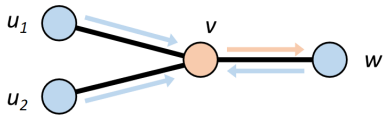


Figure 2: Directed message passing procedure.

Algorithm 1 describes the CMPNN embedding generation process. The input of the Algorithm 1 is the directed molecular graph, $G = (V, E)$, including features for all nodes $\mathbf{x}_v$, $\forall v \in V$, and features for all edges $\mathbf{x}_{e_{v,w}}, \mathbf{e}_{v,w} \in E$. The node and edge attributes are propagated at each iteration as below. In the outer loop, each step proceeds as follows, where k denotes the current depth of the message passing: First, each node $v \in V$ aggregates representations of their incoming edges in G, $\{\mathbf{h}^{k-1}(\mathrm{e}_{v,w}), \forall w \in N(v)\}$, creating an intermediate message vector $\mathbf{m}^k(v)$ instead of using information from its neighboring nodes. Note that this is the main difference between undirected and directed graph-based message

Algorithm 1 CMPNN embedding generation algorithm

Input: Graph $G(V, E)$; depth K; input node and edge features $\{\mathbf{x}_{e_{v,w}}, \forall \mathrm{e}_{v,w} \in E, \mathbf{x}_v, \forall v \in V\}$; aggregate function AGGREGATE; communicate function COMMUNICATE; weight matrix $\mathbf{W}$.
Output: Graph-wise vector representation z.

$\mathbf{h}^0(\mathrm{e}_{v,w}) \leftarrow \mathbf{x}_{e_{v,w}}, \forall \mathrm{e}_{v,w} \in E$; $\mathbf{h}^0(v) \leftarrow \mathbf{x}_v, \forall v \in V$
for $k = 1 \ldots K$ **do**
  for $v \in V$ **do**
    $\mathbf{m}^k(v) \leftarrow$ AGGREGATE$(\{\mathbf{h}^{k-1}(\mathrm{e}_{u,v}), \forall u \in N(v)\})$;
    $\mathbf{h}^k(v) \leftarrow$ COMMUNICATE$(\mathbf{m}^k(v), \mathbf{h}^{k-1}(v))$
  end for
  for e $\in E$ **do**
    $\mathbf{m}^k(\mathrm{e}_{v,w}) \leftarrow \mathbf{h}^k(v)$ - $\mathbf{h}^{k-1}(\mathrm{e}_{w,v})$;
    $\mathbf{h}^k(\mathrm{e}_{v,w}) \leftarrow \sigma(\mathbf{h}^0(\mathrm{e}_{v,w})$ + $\mathbf{W} \cdot \mathbf{m}^k(\mathrm{e}_{v,w}))$
  end for
end for
$\mathbf{m}(v) \leftarrow$ AGGREGATE$(\{\mathbf{h}^K(\mathrm{e}_{u,v}), \forall u \in N(v)\})$
$\mathbf{h}(v) \leftarrow$ COMMUNICATE$(\mathbf{m}(v), \mathbf{h}^K(v), \mathbf{x}(v))))$
$\mathbf{z} \leftarrow$ Readout$(\{\mathbf{h}(v), \forall v \in V\})$

passing. After obtaining the message vector, CMPNN then concatenates the node's current hidden state $\mathbf{h}^{k-1}(v)$ with the message vector, and this concatenated feature vector is fed through a communicative function to update the node's hidden state to $\mathbf{h}^k(v)$. The hidden state $\mathbf{h}^k(v)$ can be thought of as a message transfer station that receives the incoming messages and sends an integrated one to the next station. The communicative function can be implemented by a selection of architectures, and we discuss different architecture choices in Section 3.3 below. During the intermediate message vector generation, we also introduce an new message booster function to amplify the incoming information and the details will be further discussed in Section 3.2.

In original DMPNN, the edge message $\mathbf{m}^k(\mathrm{e}_{v,w})$ is generated based on the neighboring edge hidden states $\{\mathbf{h}^{k-1}(\mathrm{e}_{u,v}), \forall u \in N(v) \backslash w\}$. In particular, message $\mathbf{m}^k(\mathrm{e}_{v,w})$ does not depend on its inverse edge features $\mathbf{h}^{k-1}(\mathrm{e}_{w,v})$. In CMPNN, however, we have obtained a high level neighboring edges information in $\mathbf{h}^k(v)$. Thus, we can obtain the $\mathbf{m}^k(v, w)$ by subtracting its inverse bond information from the $\mathbf{h}^k(v)$. This step enables the message passing from the source node to the directed edge. For the updating of edge hidden states, $\mathbf{m}^k(\mathrm{e}_{v,w})$ is first fed into a fully connected layer and added with the initial $\mathbf{h}^0(\mathrm{e}_{v,w})$ as a skip connection following [Yang *et al.*2019], and is then transformed by a rectified linear unit (ReLU) to be used at the next iteration. The directed message passing procedure in CMPNN can be referred to Figure 2.

After iterating K steps, one more round of interaction is employed to interact the enriched bond messages and atom messages. Here, the messages from incoming bonds, current atom's representation, and the atom's initial information are gathered to obtain the final atom representation $\mathbf{h}(v)$ of the molecule through a communicative function.

Finally, a readout operator is applied to get a fixed feature vector for the molecule. We simplified the one in MPNN as:

$$\mathbf{z} = \sum_{v \in V} GRU(H(v)) \tag{1}$$

where $H(v)$ is the set of atom representations in the molecular graph G, GRU is the Gated Recurrent Unit introduced in Cho et al. (2014). Finally, we perform downstream property predictions $\hat{y} = f(h)$ where $f(\cdot)$ is a fully connected layer.

3.2 Message Booster

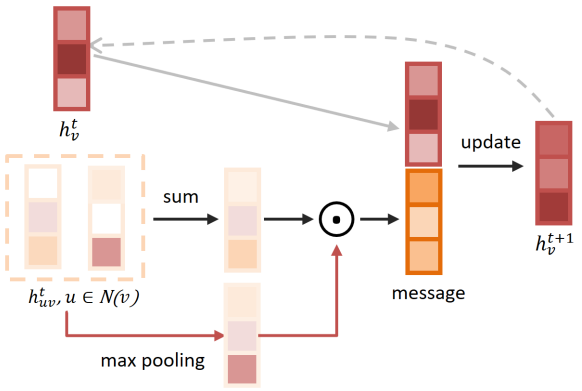


Figure 3: The architecture of the message booster.

The key step of message passing is the message aggregation. [Hamilton *et al.*2017] mentioned two efficient ways to aggregate messages from different edges, including max pooling and long short-term memory(LSTM) aggregators. [Xu *et al.*2018] demonstrated that the sum pooling aggregator could capture the full multiset and outperformed max or mean pooling. [Yang *et al.*2019] also summed the messages from their neighboring edges in updating module. These aggregators are parameter efficient but consider each edge independently without taking into account the relationships between the edges. Here, we introduced a message booster, which can be defined as:

$$\mathbf{m}_i^k(v) = \sum_{u \in N(v)} \mathbf{h}^{k-1}(e_{u,v}) \tag{2}$$

$$booster^k(v) = pooling(\mathbf{m}_i^k(v)) \tag{3}$$

$$\mathbf{m}^k(v) = \mathbf{m}_i^k(v) \odot booster^k(v) \tag{4}$$

where the $\mathbf{m}_i^k(v)$ is an intermediate message vector, and *pooling* operator is a max pooling function here. $\odot$ is an element-wise multiplication operator.

The intuition behind the message booster is that different edge messages have different strengths, and the hidden state of a node is largely on the strongest message from the incoming edges. To this end, we applied a max pooling to highlight the the edge with the highest information intensity. Other pooling strategies could also be used in this place and the results of self-attentive pooling [Veličković *et al.*2017] were provided in Section 4.2 for a comparison.

3.3 Node-Edge Message Communication

Another key step is to leverage the information from nodes and edges that is transformed to be used at next iteration. In MPNN and DMPNN, it was referred to as the updating step, since they updated the hidden state on the node or edge information. Because CMPNN updates the hidden states by interacting the node and directed edge information, this update function must have the capability to capture the interactions between node and edge features.

Here, we examined three candidate node-edge message communication modules:

Inner Product Kernel. A simple idea to interact node message and edge message is to multiply their features. The updated message can be obtained as:

$$\mathbf{h}^k(v) = \mathbf{h}^{k-1}(v) \odot \mathbf{m}(v) \tag{5}$$

where $\mathbf{m}(v)$ contains the neighboring incoming edge messages and $\mathbf{h}^{k-1}(v)$ represents the current node message. $\odot$ is an element-wise multiplication operator.

Gated Graph Kernel. The second candidate communicative function we implemented is the update function introduced in [Li *et al.*2015], as:

$$\mathbf{h}^k(v) = GRU(\mathbf{h}^{k-1}(v), \mathbf{m}(v)) \tag{6}$$

where GRU is the Gated Recurrent Unit. Compared to the inner product kernel, GRU kernel has the advantage of larger expressive capability. However, GRUs are not symmetric, as they process their inputs in a sequential way [Hamilton *et al.*2017], while the hidden vectors obtained in a molecular graph are inherently unordered.

Multilayer Perception. Another simple but useful way to incorporate both the node and edge features is to feed them into a multilayer perception. By this procedure, messages in different dimensions of the feature vectors could be interacted. It can be formulated as below:

$$\mathbf{h}^k(v) = \sigma(\mathbf{W}^k \cdot CONCAT(\mathbf{h}^{k-1}(v), \mathbf{m}(v))) \tag{7}$$

Though several other update functions were mentioned in [Gilmer *et al.*2017] [Li *et al.*2019], we utilized these three representative and classical update functions in consideration of the computation cost and complexity.

4 Experiments

4.1 Experiment Setups

Benchmark Datasets. To enable head-to-head comparison of CMPNN to existing molecular representation methods, we evaluated our proposed model on six public benchmark

Dataset	#Tasks	Task Type	#Molecule
BBBP	1	Classification	2,039
Tox21	12	Classification	7,831
Sider	27	Classification	1,427
ClinTox	2	Classification	1,478
ESOL	1	Regression	1,128
FreeSolv	1	Regression	642

Table 2: Statistics of datasets.

Task	Classification(ROC-AUC)				Regression(RMSE)	
Dataset	BBBP	Tox21	Sider	ClinTox	ESOL	FreeSolv
RF	0.788	0.619	0.572	0.544	1.176	2.048
FNN	0.899	0.788	0.652	0.688	2.152	3.043
GCN	0.690	0.829	0.638	0.807	0.970	1.40
Weave	0.671	0.820	0.581	0.832	0.610	1.220
RGAT	0.875	0.821	0.621	0.841	0.731	1.338
N-Gram	0.890	0.842	-	0.870	0.718	1.371
MPNN	0.910 ± 0.032	0.844 ± 0.014	0.641 ± 0.014	0.881 ± 0.037	0.702 ± 0.042	1.242 ± 0.249
DMPNN	0.917 ± 0.037	0.854 ± 0.012	0.658 ± 0.020	0.897 ± 0.042	0.587 ± 0.060	1.009 ± 0.207
CMPNN-IP	0.955 ± 0.007	0.848 ± 0.005	0.652 ± 0.007	0.910 ± 0.016	0.260 ± 0.011	0.870 ± 0.150
CMPNN-GG	0.955 ± 0.009	0.847 ± 0.005	0.654 ± 0.003	0.920 ± 0.016	0.263 ± 0.012	0.970 ± 0.178
CMPNN-MLP	$\mathbf{0.963 \pm 0.003}$	$\mathbf{0.856 \pm 0.007}$	$\mathbf{0.666 \pm 0.007}$	$\mathbf{0.933 \pm 0.012}$	$\mathbf{0.233 \pm 0.015}$	$\mathbf{0.819 \pm 0.147}$

Table 3: Prediction results of CMPNN, its variants and baselines on six chemical graph datasets. We used a 5-fold cross validation with random split and replicated experiments on each tasks for five times, and reported the mean and standard deviation of AUC or RMSE values.

datasets, including BBBP, Tox21, ClinTox, and Sider for classification tasks, and ESOL and Freesolv for regression tasks.

The blood-brain barrier penetration (**BBBP**) dataset includes binary labels for over 2040 compounds on their permeability properties. The **Tox21** dataset was originally used in the Tox21 data challenge, which contains 7831 experimental compounds for 12 different targets relevant to drug toxicity, including stress response pathways and nuclear receptors. The **Sider** data set provides information on marketed drugs and their corresponding adverse drug reactions, where the side effects of 1427 approved drugs have been grouped into 27 system organ classes. The **ClinTox** dataset includes 1491 drug molecules that approved through FDA and a list of compounds that failed during clinical trials due to the toxicity. The **ESOL** dataset consists of 1128 compounds and their corresponding water solubility in log10(mol/L). The Free Solvation Database (**FreeSolv**) comprises experimental and calculated hydration free energy for small neutral molecules in water. It includes totally 642 molecules which are mostly fragment-like. The statistics of datasets are shown in Table 2.

Baselines. We compared our CMPNN with 9 baseline methods. These methods were shown in the MoleculeNet [Wu *et al.*2018] , DMPNN [Yang *et al.*2019], and several state-of-the-arts methods. The Random Forests (RF) [Breiman2001] is one of the most common used machine learning methods. The input of RF in experiments is the binary Morgan fingerprints. The FNN is a feed-forward network that also uses molecular fingerprint features as inputs. Besides, we compared our method with four graph models. GCN [Kipf and Welling2016] and Weave [Kearnes *et al.*2016] are two graph convolutional methods by adding edge attributes as node's feature. N-Gram [Liu *et al.*2019] is a state-of-the-art unsupervised representation method for molecular property prediction. RGAT [Ryu *et al.*2018] is an improved molecular graph neural network by incorporating attention and gate mechanisms. MPNN [Gilmer *et al.*2017] and DMPNN [Yang *et al.*2019] are two recent message passing methods operated on undirected and directed graph, respectively.

Implementation Details. Following [Yang *et al.*2019], we used a 5-fold cross validation and replicate experiments on each task for five times, and reported the mean and standard deviation of AUC or RMSE values. We evaluated all models on random and scaffold-based splits as recommended by [Wu *et al.*2018]. Scaffold Splitting is a more challenging and realistic evaluation setting by guaranteeing the Murcko scaffold diversity of the training validation and test sets. The node and edge features used in this paper were listed in Supplementary Information, which are computed by open-source package RDKit. To improve model performance, we applied the Bayesian Optimization to obtain the best hyperparameters of the models. Our models were implemented by Pytorch and run on Ubuntu Linux 16 with NVIDIA Tesla V100 GPUs.

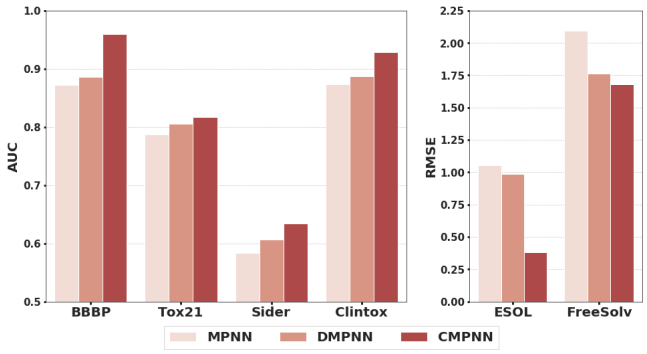
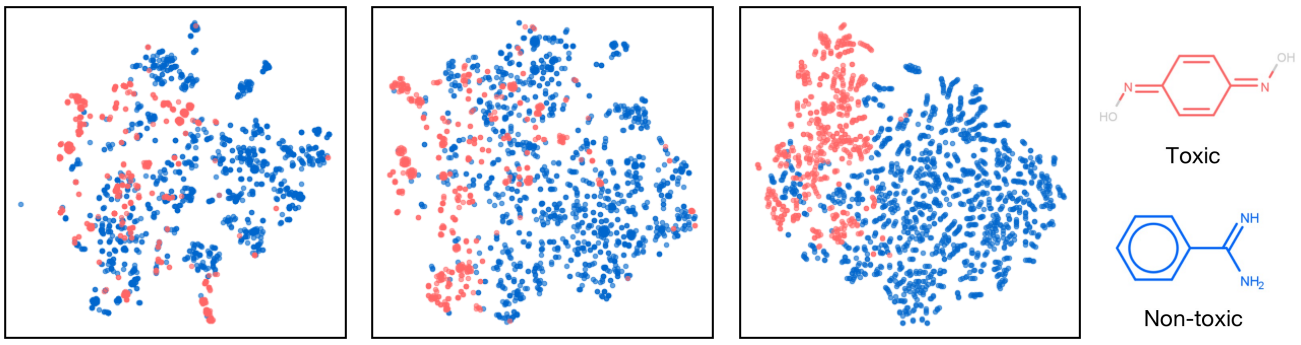


Figure 4: Comparison of CMPNN against baseline models on chemical graph datasets on a scaffold data split.

4.2 Performance Comparasion

Performance in Graph Classifications. Table 3 shows the AUC results of seven different baseline models on four classification datasets. The Tox21, Sider and ClinTox are all multiple-task learning tasks, including totally 42 classification tasks. For our CMPNN model, we also implemented three variants on the communication function as described in Section 3.3. For notational convenience we used CMPNN-IP to denote Inner Product kernel, CMPNN-GG for Gated Graph kernel and CMPNN-MLP for Multilayer Perception kernel.

Compared to traditional baselines and several primitive graph neural networks, MPNN achieved large increases of



(a) MPNN (b) DMPNN (c) CMPNN

Figure 5: t-SNE visualization of atom features for Tox21 subtask extracted from (a) MPNN, (b) DMPNN and (c) CMPNN, where atoms with toxicity are colored in red and atoms in non-toxic molecules are in blue.

AUCs in almost all datasets, except that it was slightly worse than FNN in the Sider dataset, while the DMPNN consistently improved over MPNN in four datasets by 0.8-2.7% with the use of directed message passing. By using the enhanced node-edge information interactions and message booster modules, our CMPNN-mlp further improved over DMPNN with 4.6%, 3.6% and 2.5% for BBBP, ClinTox, and Sider datasets, respectively.

Generalization Estimation. Over the Tox21 dataset, CMPNN was only marginally better than DMPNN in Tox21. This is likely because Tox21 is an extremely imbalanced dataset, where only 7% of data points are labeled as toxic compounds. [Mayr et al.2018] reported that this kind of dataset is prone to be over-fitting and thus may perform worse in the independent test dataset. For this reason, we introduced Scaffold Splitting to further evaluate the generalization ability of different MPNN variants. As shown in Figure 4, the CMPNN achieved an order-of-magnitude improvement over MPNN and DMPNN with a scaffold splitting strategy. In Tox21 task, our CMPNN model improves upon GCN by a margin of 1.1% for the test set. This result demonstrates that the CMPNN has a better generalization ability than the previous MPNNs when the training data set has no similar scaffold to the test set.

Performance in Graph Regressions. Solubility are one kind of basic physical chemistry property important for understanding how molecules interact with solvents. Table 3 compares CMPNN results to other state-of-the-art models results on two solubility datasets. The best-case RMSE of the CMPNN model on ESOL and FreeSolv were 0.233 $\pm$ 0.015 log M and 0.819 $\pm$ 0.147 kcal/mol, which improved upon DMPNN by a margin of 0.354 logM and 0.190 kcal/mol with the same fold assignments, respectively. These results indicate that better representations of molecular graphs could be obtained by the updating of both the vertices and edges messages in CMPNN during training.

Task	Classification	Regression
Dataset	BBBP	ESOL
Without All	0.925	0.453
Without Communicate	0.937	0.361
Without Booster	0.951	0.267
Booster-Attentive	0.956	0.254
CMPNN	**0.963**	**0.233**

Table 4: Ablation results on BBBP and ESOL datasets.

4.3 Ablation Study

We conducted ablation studies on the two benchmarks to investigate factors that influence the performance of proposed CMPNN framework. As shown in Table 4, CMPNN with the max pooling booster and communicating modules shows the best performance among all architectures. The exclusion of the readout function in the "without ALL" variant performed the worst. The exclusions of the message booster and node-edge message communicative function both caused large decreases in performances. The use of attentive pooling as the booster [Veličković *et al.*2017] is helpful but not as efficient as max pooling.

4.4 Atomic Representation Visualization.

In chemistry, molecular properties are often associated with their specific substructures. Thus, recognizing substructures related to the target property is important to achieve a high performance. In this regard, we compared the learning capabilities of three methods in the atomic level. As an example of the subtask SR-MMP in Tox21, we selected 100 toxic molecules containing substructures matched with the PAINS database [Baell and Holloway2010] (a database containing more than 400 toxic substructures), and took the matched substructure atoms as toxic. In control, we selected 100 non-toxic molecules and took the atoms as non-toxic. Finally, we obtained 564 toxic and 1367 non-toxic atoms. Figure 5 shows the toxic (red) and non-toxic atoms (blue) by projecting their learned atomic feature vectors by the t-distributed stochastic

neighbor embedding (t-SNE) with default hyper-parameters. Overall, all methods provid reasonable results. While one portion of toxic atoms represented by MPNN and DMPNN are mixed with non-toxic atoms, CMPNN allows more delicate classifications. This result suggests that CMPNN capture better representations of molecules.

5 Conclusions

In this paper, we propose a directed graph-based Communicative Message Passing Neural Network (CMPNN) to improve the molecular embedding by strengthening the message interaction between atoms and bonds. A message booster module and a communicative function are introduced to support the message propagation process. Extensive experiments demonstrate that our CMPNN obtains superior performances against state-of-the-art baselines on six chemical graph datasets.

Acknowledgments

The work was supported in part by the National Key RD Program of China (2018YFC0910500), GD Frontier and Key Tech Innovation Program (2018B010109006,2019B020228001), Shenzhen Science and Technology Innovation Council (Grant No. JCYJ20180508162601910), the National Natural Science Foundation of China (61772566, U1611261 and 81801132, 81903540, 62041209) and the programme for Guangdong Introducing Innovative and Entrepreneurial Teams (2016ZT06D211).

References

[Baell and Holloway, 2010] Jonathan B Baell and Georgina A Holloway. New substructure filters for removal of pan assay interference compounds (pains) from screening libraries and for their exclusion in bioassays. *Journal of medicinal chemistry*, 53(7):2719–2740, 2010.

[Breiman, 2001] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Cherkasov *et al.*, 2014] Artem Cherkasov, Eugene N Muratov, Denis Fourches, Alexandre Varnek, Igor I Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C Martin, Roberto Todeschini, et al. Qsar modeling: where have you been? where are you going to? *Journal of medicinal chemistry*, 57(12):4977–5010, 2014.

[Coley *et al.*, 2017] Connor W Coley, Regina Barzilay, William H Green, Tommi S Jaakkola, and Klavs F Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling*, 57(8):1757–1772, 2017.

[Dahl *et al.*, 2014] George E Dahl, Navdeep Jaitly, and Ruslan Salakhutdinov. Multi-task neural networks for qsar predictions. *arXiv preprint arXiv:1406.1231*, 2014.

[Duvenaud *et al.*, 2015] 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 *Advances in neural information processing systems*, pages 2224–2232, 2015.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[Hamilton *et al.*, 2017] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. *arXiv e-prints*, page arXiv:1706.02216, Jun 2017.

[Jastrzebski *et al.*, 2016] Stanisław Jastrzebski, Damian Leśniak, and Wojciech Marian Czarnecki. Learning to smile (s). *arXiv preprint arXiv:1602.06289*, 2016.

[Kearnes *et al.*, 2016] 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.

[Kipf and Welling, 2016] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[Li *et al.*, 2015] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[Li *et al.*, 2019] Ziyao Li, Liang Zhang, and Guojie Song. Gcn-lase: towards adequately incorporating link attributes in graph convolutional networks. *arXiv preprint arXiv:1902.09817*, 2019.

[Liu *et al.*, 2019] Shengchao Liu, Mehmet F Demirel, and Yingyu Liang. N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. In *Advances in Neural Information Processing Systems*, pages 8464–8476, 2019.

[Nettles *et al.*, 2007] James H Nettles, Jeremy L Jenkins, Chris Williams, Alex M Clark, Andreas Bender, Zhan Deng, John W Davies, and Meir Glick. Flexible 3d pharmacophores as descriptors of dynamic biological space. *Journal of Molecular Graphics and Modelling*, 26(3):622–633, 2007.

[Rogers and Hahn, 2010] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[Ryu *et al.*, 2018] Seongok Ryu, Jaechang Lim, Seung Hwan Hong, and Woo Youn Kim. Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network. *arXiv preprint arXiv:1805.10988*, 2018.

[Shang *et al.*, 2018] Chao Shang, Qinqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, Jinfeng Yi, and Jinbo Bi. Edge attention-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1802.04944*, 2018.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2017.

[Withnall *et al.*, 2020] M Withnall, E Lindelöf, O Engkvist, and H Chen. Building attention and edge message passing neural networks for bioactivity and physical–chemical property prediction. *Journal of Cheminformatics*, 12(1):1, 2020.

[Wu *et al.*, 2018] 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.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[Yang *et al.*, 2019] Kevin Yang, Kyle Swanson, Wengong Jin, Connor Coley, Philipp Eiden, Hua Gao, Angel Guzman-Perez, Timothy Hopper, Brian Kelley, Miriam Mathea, et al. Are learned molecular representations ready for prime time? *arXiv preprint arXiv:1904.01561*, 2019.

[Zheng *et al.*, 2019a] Shuangjia Zheng, Jiahua Rao, Zhongyue Zhang, Jun Xu, and Yuedong Yang. Predicting retrosynthetic reactions using self-corrected transformer neural networks. *Journal of Chemical Information and Modeling*, 2019.

[Zheng *et al.*, 2019b] Shuangjia Zheng, Xin Yan, Yuedong Yang, and Jun Xu. Identifying structure–property relationships through smiles syntax analysis with self-attention mechanism. *Journal of chemical information and modeling*, 59(2):914–923, 2019.

[Zheng *et al.*, 2020] Shuangjia Zheng, Yongjian Li, Sheng Chen, Jun Xu, and Yuedong Yang. Predicting drug–protein interaction using quasi-visual question answering system. *Nature Machine Intelligence*, 2(2):134–140, 2020.