Learning Attributed Graph Representations with Communicative Message Passing Transformer

Jianwen Chen[1†], **Shuangjia Zheng**[1,4†*], **Ying Song**[2], **Jiahua Rao**[1,4] and **Yuedong Yang**[1,3*]
[1]School of Computer Science and Engineering, Sun Yat-sen University
[2]School of Systems Science and Engineering, Sun Yat-sen University
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Sun Yat-sen University
[4]Galixir Technologies Ltd, Beijing
{chenjw48, zhengshj9, songy75, raojh6}@mail2.sysu.edu.cn, yangyd25@mail.sysu.edu.cn

Abstract

Constructing appropriate representations of molecules lies at the core of numerous tasks such as material science, chemistry and drug designs. Recent researches abstract molecules as attributed graphs and employ graph neural networks (GNN) for molecular representation learning, which have made remarkable achievements in molecular graph modeling. Albeit powerful, current models either are based on local aggregation operations and thus miss higher-order graph properties or focus on only node information without fully using the edge information. For this sake, we propose a Communicative Message Passing Transformer (CoMPT) neural network to improve the molecular graph representation by reinforcing message interactions between nodes and edges based on the Transformer architecture. Unlike the previous transformer-style GNNs that treat molecules as fully connected graphs, we introduce a message diffusion mechanism to leverage the graph connectivity inductive bias and reduce the message enrichment explosion. Extensive experiments demonstrated that the proposed model obtained superior performances (around 4% on average) against state-of-the-art baselines on seven chemical property datasets (graph-level tasks) and two chemical shift datasets (node-level tasks). Further visualization studies also indicated a better representation capacity achieved by our model.

1 Introduction

Accurate characterization of molecular properties remains largely an open challenge, and its solution may unlock a widespread use of deep learning in the drug discovery industry [Wu *et al.*, 2018]. Traditionally, this has involved translating a molecule m to a dense feature vector with a representation function, $h = g(m)$, and then applying a variety of techniques to predict the targeted property based on the representation by $y = f(h)$.

Early predictive modeling methods such as quantitative structure-property relationships (QSPR) have been performed based on fixed representations such as expert-crafted physicochemical descriptors and molecular fingerprints [Rogers and Hahn, 2010]. However, descriptor-based methods presume that all target property-related information is covered by the chosen descriptor set, limiting the capability for a model to make problem-specific decisions.

More naturally, a molecular structure can be abstracted as a topological graph with attributed nodes and edges, where node features correspond to atom properties like atomic identity and degree, edge features correspond to bond properties, like bond type and aromaticity. In this sense, graph representation models, especially Graph Neural Networks (GNN), can be intuitively introduced to learn the representations of molecules. Generally, the procedure of GNN framework can be summarized in three main steps: (1) Initialization step, where nodes are initialized with their initial attributes or structural features; (2) Message Passing step, where the features at each node are transmitted from its neighbors across the molecular graph into a message vector; (3) Read-out step, where the node messages are aggregated or pooled into a fixed-length feature vector. Under the above framework, many GNN architectures have been proposed for effective graph representation learning, that achieve promising results in many property prediction tasks [Duvenaud *et al.*, 2015; Yang *et al.*, 2019; Song *et al.*, 2020].

Despite the fruitful progress, several issues still impede the performance of the current GNN in the molecular graph. First, common graph convolutional operations aggregate only local information and suffer from the suspended animation problem when stacking excessive GNN layers [Zhang and Meng, 2019], so these models are naturally difficult to learn long-range dependencies and the global chemical environment of each atom. Second, main-stream GNN and its variants mainly focus on obtaining effective nodes embedding but weaken the information carried by edges that is also important for informative graph representations [Shang *et al.*, 2018]. Meanwhile, the node representations obtained by such deep models tend to be over-smoothed and hard to distinguish [Li *et al.*, 2018]. Such issues greatly hinder the applications of GNNs for molecular representation learning tasks.

[†]These two authors contributed equally.
[*]Corresponding authors.
[‡]https://github.com/jcchan23/CoMPT

To address the above problems, many efforts have been made from different directions. On the one hand, with the emerging of Transformer [Vaswani *et al.*, 2017] in sequence modeling, several Transformer-style GNNs [Chen *et al.*, 2019; Maziarka *et al.*, 2020] have been introduced to learn the long-range dependencies in graph-structured data. These methods can be viewed as a variant of the Graph Attention Network (GAT) [Veličković *et al.*, 2017] on a fully connected graph constructed by all atoms, which ignore the graph connectivity inductive bias. As a result, they perform poorly in the tasks where graph topology plays an important rule. On the other hand, the directed message passing neural network [Yang *et al.*, 2019] and its variants [Song *et al.*, 2020] have been proposed to transmit messages through directed edges rather than vertices. Such methods make use of the edge information explicitly and avoid unnecessary loops in the message passing trajectory, but they still cannot deal with the long-range dependencies.

Based on these observations, we propose a Communicative Message Passing Transformer (CoMPT) neural network for molecular representation learning. In contrast to the previous Transformer-style GNNs that emphasize the node information, CoMPT invokes a communicative message-passing paradigm by strengthening the message interactions between edges and nodes. In our framework, both the edge and node embeddings are updated during the training process. Besides, we refine the message passing process by using the topological connection matrix with a diffusion mechanism to reduce the message enrichment explosion. By selectively propagating information within a molecular graph, CoMPT is able to extract more expressive representation for down-stream tasks. The main contributions of this work include:

- We propose a novel communicative message passing transformer, namely CoMPT, that explicitly captures the atom and bond information of molecular graphes and incorporates both local and global structural information.
- Our model includes an elegant way to fuse topology connection matrix using the message diffusion mechanism inspired by the thermal diffusion phenomenon, which was further demonstrated to alleviate the oversmoothing problem.
- Numerical experiments are conducted on both graph-level and node-level public datasets to demonstrate the effectiveness of our method. CoMPT surpasses the state-of-the art models on all nine tasks by up-to 4% improvement in the average performance.

2 Related Work

Molecular representation learning. One of the most popular representations of molecules is the fixed representations through chemical fingerprints, such as Extended Connectivity Fingerprints (ECFP) [Rogers and Hahn, 2010] and chemical descriptors. The heuristics integrated in descriptor generation algorithms typically embed high-level chemistry principles, attempting to maximize the information content of the resulting feature vectors. While these methods can be clearly successful, they always feature a trade-off by emphasizing certain molecular features, while neglecting others. The selections of features are hard-coded in the algorithm and not amenable to problem-specific tuning. Recent works started to explore the molecular graph representation. Early studies learned to only encode the node features [Duvenaud *et al.*, 2015] without considering bond information. To gain supplementary information from edges, [Kearnes *et al.*, 2016] proposed to utilize attributes of both atoms and bonds, and [Gilmer *et al.*, 2017] summarized it into a MPNN framework. Though a few more studies used the information of the edges through network modules such as the edge memory module [Withnall *et al.*, 2020], these models were mainly built upon the node-based MPNN and thus still suffered from the information redundancy during message aggregations. DMPNN [Yang *et al.*, 2019] was introduced as an alternative as it abstracted the molecular graph as an edge-oriented directed graph, avoiding the unnecessary loops in message passing procedure. CMPNN [Song *et al.*, 2020] further extend this work by strengthening the message interactions between nodes and edges through a communicative kernel. Our work is closely related to CMPNN, while our model built on Transformer is more elegant to capture long-range dependencies and structural variety.

Transformer-style graph neural network. Several attempts have been made to integrate transformer and graph neural network. [Chen *et al.*, 2019] introduced the Path-Augmented Graph Transformer Networks to explicitly take account for longer-range dependencies in molecular graph. One closely related work is [Maziarka *et al.*, 2020], which proposed a Molecule Attention Transformer by augmenting the attention mechanism in Transformer using inter-atomic distances and the molecular graph structure. Another work worth to note is GROVER [Rong *et al.*, 2020], which provided a self-supervised pre-trained transformer-style message passing neural network for molecular representation learning. While our model also builds on Transformer [Vaswani *et al.*, 2017] to encode graphs, we contribute new techniques to leverage the graph connectivity inductive bias.

3 Methods

In this section, we first briefly review basic concepts of the Transformer model [Vaswani *et al.*, 2017]. Then, we focus on our contributions, describing our alternative in the Transformer encoder framework that uses node-edge message interaction module instead of the self-attention mechanism to pass the message and learn expressive representations for attributed molecular graphs. Finally, we introduce a message diffusion mechanism for leveraging the graph connectivity inductive bias and reducing the message enrichment explosion.

3.1 Preliminary

Notation and problem definition. A molecular structure can be considered as an attributed graph $G = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = n$ denotes a set of n atoms (nodes) and $|\mathcal{E}| = m$ denotes a set of m bonds (edges). $\mathcal{N}_v$ is utilized to denote the set of node v's neighbors. For each node and edge, we use f_{node} and f_{edge} represents the feature dimensions, respectively. Following [Yang *et al.*, 2019] that passing message

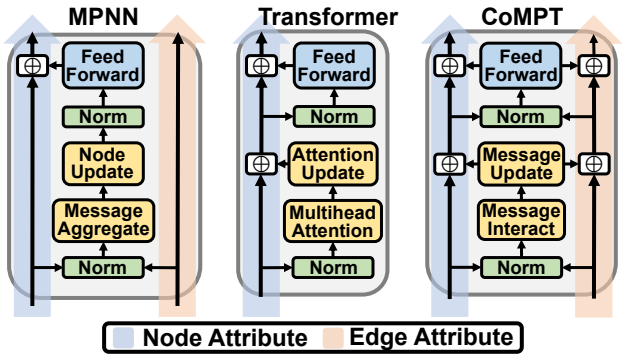

Figure 1: Comparing message passing procedure among MPNN (left), Transformer (middle) and CoMPT (right).

with directed edges, we treat molecular structures as directed graphs to avoid messages being passed along any unnecessary loops in the aggregation procedure. As such, we use $x_v \in \mathbb{R}^{f_{node}}$ to represent the initial features of node v, and $e_{uv} \in \mathbb{R}^{f_{edge}}$ are the initial features of the edge (u, v) with direction $u \to v$. $X \in \mathbb{R}^{n \times f_{node}}$ and $E \in \mathbb{R}^{n \times n \times f_{edge}}$ are the matrices form for all nodes and edges. Besides, there are generally two categories of supervised tasks in the molecular graph learning problems: i) *Graph classification/regression*, where a set of molecular graphs $\{G_1, \ldots, G_N\}$ and their labels/targets $\{y_1, \ldots, y_N\}$ are given, and the task is to predict the label/target of a new graph. ii) *Node classification/regression*, where each node v in multiple graphs has a label/target y_v, and the task is to predict the labels/targets of nodes in the unseen graphs.

Attention mechanism. Our CoMPT model are built on the transformer encoder framework, in which the attention module is the main building block. The usual implementation of the attention module is the dot product self-attention, which takes inputs with a set of queries, keys, and values (q, k, v) that are projected from hidden node features $h(X)$. Then it computes the dot product of the query with all keys and applies a *softmax* function to obtain weights on the values. By stacking the set of (q, k, v) s into matrices (Q, K, V), it allows highly optimized matrix multiplication operations. Specifically, the outputs can be formulated as:

$$\begin{cases} [Q, K, V] = h(X)[W^Q, W^K, W^V] \\ Attention(Q, K, V) = softmax(QK^T/\sqrt{d})V \end{cases} \tag{1}$$

where d is the dimension of q and k. Furthermore, we focus on the multi-head attention, where l attention layers are stacked together. The output matrix can be extended as,

$$\begin{cases} Multihead(Q, K, V) = Cat(head_1, \ldots, head_l)W^O \\ head_i = Attention(h(X)W_i^Q, h(X)W_i^K, h(X)W_i^V) \end{cases} \tag{2}$$

where W_i^Q, W_i^K, W_i^V are the projection matrices of head i.

3.2 The Framework of CoMPT

Encoding for node position and edge direction. Compared to the Transformer encoder that only takes node attributes, our CoMPT takes three inputs: node features X, edge features E, and the topology connection matrix $A \in \mathbb{R}^{n \times n}$ that is computed by using the length of the shortest path between any two nodes. Since the initial node and edge features (more details are listed in the Appendix) do not involve information related to the node position and the edge direction, we need to add annotations explicitly with these features before being fed into the CoMPT model. Specifically, for any node $v_i (i = 1, 2, \ldots, n)$ and its corresponding initial feature x_i, we train a learnable position embedding vector pos_i according to the atomic index of the node, and then add the initial features to get the hidden features, which can be formulated as:

$$h(x_i) = node\_embedding(x_i) + pos_i \tag{3}$$

where $node\_embedding()$ projects the initial feature to the corresponding dimension. For any directed edge e_{uv}, we followed the previous method in [Song *et al.*, 2020] by adding the source node feature to the initial feature. It could be formulated as:

$$h(e_{uv}) = edge\_embedding(e_{uv}) + h(x_u) \tag{4}$$

where $edge\_embedding()$ also projects the initial feature to the corresponding dimension. For convenience, we set the same dimension f for all hidden features.

Node-Edge message interaction module. The key idea behind CoMPT is that we use hidden node features $h(X)$ and hidden edge features $h(E)$ to compute a message interaction scores M, which replaces the self-attention scores in the original encoder layer. Specifically, three matrices Q, K, V are firstly calculated by the formula:

$$\begin{cases} [Q, V] = h(X)[W^Q, W^V] \\ K = h(E)W^K \end{cases} \tag{5}$$

where W^Q, W^K, W^V are the projection matrices. The message interaction matrix T is generated by the inner product:

$$T = matmul(Q, K.tranpose(-2, -1)) \tag{6}$$

or equivalently in each position

$$T[i, u, v] = matmul(q_i, k_{uv}) \tag{7}$$

where q_i and k_{uv} denote the hidden vector of node i and directed edge (u, v), respectively. The intuition behind this tensor product is straight-forward: we compute the scalar product between each node and edge in order to generate the structural message related to the molecule for the final prediction. Subsequently, a selection step is applied to the message interaction matrix T to preserve the molecular graph connectivity. Here, we select three types of matrices according to the node's neighbors: the node interacts with its outgoing edges ($\boldsymbol{M_o}$), incoming edges ($\boldsymbol{M_i}$) and the self-loop edge ($\boldsymbol{M_d}$). For convenience, the generation and the selection step could be merged by *einsum* operation with the formula below:

$$\begin{cases} M_o = einsum_{nf,nmf \to nm}(Q, K) \\ M_i = einsum_{nf,mnf \to nm}(Q, K) \\ M_d = diag(M_o) = diag(M_i) \end{cases} \tag{8}$$

After computing three matrices, we normalize them to sum up to 1.0 in each row with the *softmax* function σ, and compute the final message by:

$$M = \sigma(M_o) + \sigma(M_i) - \sigma(M_d) \tag{9}$$

This operation eliminates the double-counted self-loop message and avoids the information explosion. This final message is further utilized to update the node hidden features $h(X)$ and edge hidden features $h(E)$ in each encoder layer. Furthermore, in the situation of multi-head attention, similar to the original transformer framework, we also stack all l attention blocks at the end of each layer.

Residual message update module. In the original transformer encoder framework, the self-attention scores are utilized to compute the weighted sum of vectors for each node, which could be regarded as an updated operation that is applied to all nodes. In contrast, we utilize the message interaction scores M to update the node hidden features $h(X)$. Besides, inspired by [Song *et al.*, 2020], the edge hidden features are also updated according to the rich information that comes from M. The update operation is concluded as:

$$\begin{cases} h(X) = matmul(M, V) \\ h(E) = M \odot K \end{cases} \tag{10}$$

where $\odot$ denotes the element-wise operation. Besides, CoMPT model has multiple stacked layers, where each encoder layer consists of a multi-head message interaction module, message update module and a position-wise forward module. To make the training step more stable, we adopt the post layer norm module [Xiong *et al.*, 2020] before getting into each module. Furthermore, residual connections between any two encoder layers are added for reducing the vanishing of the gradient, which can be formulated as:

$$\begin{cases} h_{k+1}(X) = h_k(X) + Encoder(h_k(X), h_k(E)) \\ h_{k+1}(E) = h_k(E) + Encoder(h_k(X), h_k(E)) \end{cases} \tag{11}$$

where k represents the index of the encoder layer, $Encoder(\cdot)$ denotes the whole encoder layer with the various modules mentioned above.

3.3 Message Diffusion and Global Pooling

The key to accurately predict the properties on the graph-level/node-level tasks is how to keep the message interacting correctly. Previous studies have shown that deep message aggregation of GNN will lead to an over-smooth phenomenon[Li *et al.*, 2018], where the features of nodes within the graph will converge to the same values.

To alleviate this issue, we design a simple attenuation mechanism for message passing during iteration to delay the process of aggregating redundant information to nodes. In particular, each hidden vector in message interaction scores M could be regarded as the prepared message sent from the row index of the node to the column index of the node, and the topology connection matrix A shows the distance of shortest path between two nodes. We add the attenuation coefficient by using the Gaussian kernel function with the formula:

$$M(u, v) = M(u, v)e^{-\alpha A(u,v)} \tag{12}$$

where $M(u, v)$ denotes the message sending from u to v, $A(u, v)$ means the shortest path between u and v, $\alpha \in [0, 1]$ is a trainable coefficient to control the attenuation level. It is obvious that with the increase of distance, the message will decay rapidly in the beginning, and then turn smooth for a long distance. After applying this mechanism, we can defer the over-smoothness of the aggregating process to a certain degree. The ablation study in the section 4.3 also shows that the attenuation mechanism improves the prediction performance.

Finally, for the graph-level tasks, a readout operator/generation layer is added to obtain a fixed feature vector for the molecule. Here, we adopt a Gated Recurrent Unit (GRU) for global pooing following [Gilmer *et al.*, 2017; Song *et al.*, 2020] as:

$$z = \sum_{x \in \mathcal{V}} GRU(h(x)) \tag{13}$$

where $h(x)$ is the set of atom representations in the molecular graph, and GRU is the Gated Recurrent Unit. Finally, we perform downstream property prediction $\hat{y} = f(h)$ where $f(\cdot)$ is a fully connected layer.

4 Experiments

In this section, we evaluate the proposed model CoMPT on three kinds of tasks. We aim to answer the following research questions:

- **RQ1**: How does CoMPT model perform compared with state-of-the-art molecular property prediction methods?
- **RQ2**: How do different components (i.e, node position, edge direction, and message diffusion mechanism) affect CoMPT?
- **RQ3**: Can CoMPT model provide better representations for the attributed molecular graphs?

4.1 Experiment Setups

Benchmark Datasets

To enable head-to-head comparisons of CoMPT to existing molecular representation methods, we evaluated our proposed model on nine benchmark datasets across three kinds of tasks from [Wu *et al.*, 2018] and [Jonas and Kuhn, 2019], each kind of which consists of 2 to 4 public benchmark datasets, including BBBP, Tox21, Sider, and ClinTox for Graph Classification tasks, ESOL, FreeSolv and Lipophilicity for Graph Regression tasks, chemical shift prediction of hydrogen and carbon for Node Regression tasks. The statistics of datasets are shown in Table S1.

In the graph-level task, following the previous works, we utilized a 5-fold cross-validation and replicate experiments on each task five times. Note that we adopted the scaffold split method recommended by [Yang *et al.*, 2019] to split the datasets into training, validation, and test, with a 0.8/0.1/0.1 ratio. Scaffold Split is a more challenging and realistic evaluation setting in molecular property prediction tasks by guaranteeing the high molecular scaffold diversity of the training, validation, and test sets.

In the node-level task, we follow the previous study [Jonas and Kuhn, 2019] by randomly splitting the dataset into 80% as the training set and 20% as the test set, and then use 95% of training data to train the model and the remaining 5% to validate the model for early stopping. All methods report the mean and standard deviation of corresponding metrics. To improve model performance, we applied the grid search to obtain the best hyper-parameters of the models.

Task	Graph Classification(ROC-AUC)				Graph Regression(RMSE)		
Dataset	BBBP	Tox21	Sider	ClinTox	ESOL	FreeSolv	Lipophilicity
TF_Robust	0.860 ± 0.087	0.698 ± 0.012	0.607 ± 0.033	0.765 ± 0.085	1.722 ± 0.038	4.122 ± 0.085	0.909 ± 0.060
GCN	0.877 ± 0.036	0.772 ± 0.041	0.593 ± 0.035	0.845 ± 0.051	1.068 ± 0.050	2.900 ± 0.135	0.712 ± 0.049
Weave	0.837 ± 0.065	0.741 ± 0.044	0.543 ± 0.034	0.823 ± 0.023	1.158 ± 0.055	2.398 ± 0.250	0.813 ± 0.042
SchNet	0.847 ± 0.024	0.767 ± 0.025	0.545 ± 0.038	0.717 ± 0.042	1.045 ± 0.064	3.215 ± 0.755	0.909 ± 0.098
N-Gram	0.912 ± 0.013	0.769 ± 0.027	0.632 ± 0.005	0.855 ± 0.037	1.100 ± 0.160	2.512 ± 0.190	0.876 ± 0.033
AttentiveFP	0.908 ± 0.050	0.807 ± 0.020	0.605 ± 0.060	0.933 ± 0.020	0.853 ± 0.060	2.030 ± 0.420	0.650 ± 0.030
MPNN	0.913 ± 0.041	0.808 ± 0.024	0.595 ± 0.030	0.879 ± 0.054	1.167 ± 0.430	2.185 ± 0.952	0.672 ± 0.051
MGCN	0.850 ± 0.064	0.707 ± 0.016	0.552 ± 0.018	0.634 ± 0.042	1.266 ± 0.147	3.349 ± 0.097	0.650 ± 0.030
DMPNN	0.919 ± 0.030	0.826 ± 0.023	0.632 ± 0.023	0.897 ± 0.040	0.980 ± 0.258	2.177 ± 0.914	0.653 ± 0.046
CMPNN	0.927 ± 0.017	0.806 ± 0.016	0.616 ± 0.003	0.902 ± 0.008	0.798 ± 0.112	2.007 ± 0.442	0.614 ± 0.029
Smiles Transformer	0.900 ± 0.053	0.706 ± 0.021	0.559 ± 0.017	0.905 ± 0.064	1.144 ± 0.118	2.246 ± 0.237	1.169 ± 0.031
GROVER	0.911 ± 0.008	0.803 ± 0.020	0.624 ± 0.006	0.884 ± 0.013	0.911 ± 0.116	1.987 ± 0.072	0.643 ± 0.030
CoMPT	$\mathbf{0.938 \pm 0.021}$	0.809 ± 0.014	$\mathbf{0.634 \pm 0.030}$	$\mathbf{0.934 \pm 0.019}$	$\mathbf{0.774 \pm 0.058}$	$\mathbf{1.855 \pm 0.578}$	$\mathbf{0.592 \pm 0.048}$

Table 1: Prediction results of CoMPT and baselines on seven chemical graph datasets. We used a 5-fold cross validation with scaffold split and replicated experiments on each tasks for five times. Mean and standard deviation of AUC or RMSE values are reported.

Baselines Comparison

We comprehensively compared CoMPT against with 12 baseline methods in the graph level task. These models were most shown in the MoleculeNet [Wu *et al.*, 2018] and GROVER as follows: TF_Robust [Ramsundar *et al.*, 2015] is a DNN-based multitask framework taking the molecular fingerprints as the input. GCN, Weave, and SchNet [Duvenaud *et al.*, 2015; Kearnes *et al.*, 2016] are three graph convolutional models. N-Gram [Liu *et al.*, 2019] is a state-of-the-art unsupervised representation method for molecular property prediction. AttentiveFP [Xiong *et al.*, 2019] is an extension of the graph attention network. MPNN and its variants MGCN [Lu *et al.*, 2019], DMPNN and CMPNN are models considering the edge features during message passing. Specifically, to demonstrate the power of the message-interaction module, we also compare CoMPT with two transformer model: Smiles Transformer [Honda *et al.*, 2019] and GROVER. For a fair comparison, we only report the results without the pre-trained strategy.

In the node level task, we compared our CoMPT model with the other 3 proposed methods in this benchmark. The first one is HOSE codes, which attempted to summarize the neighborhood around each atom in concentric spaces, and then use a nearest-neighbor approach to predict the particular shift value. The rest baselines include GCN [Jonas and Kuhn, 2019] and MPNN [Kwon *et al.*, 2020], where they used different deep graph neural networks to improve the performance of prediction.

4.2 Performance Comparison (RQ1)

Performance in graph level task. Table 1 displays the complete results of each model on all datasets, where cells in the gray shadow denote the previous best methods, and cells with the bold style show the best result achieved by CoMPT. Table 1 presents some observations: (1) Both the message passing neural network and transformer framework perform better than graph neural network on most datasets, and CoMPT combines the advantages of them to achieve the best performances on 6 out of 7 datasets. Compared to the previous best message passing method CMPNN and transformer method GROVER, the general improvements are 3.4% (2.0% on classification tasks and 3.4% on regression tasks) and 4.7% (2.7% on classification tasks and 4.7% on regression tasks), respectively. This notable incresing suggests the effectiveness of the structural representation learned by CoMPT for graph level prediction tasks. (2) The message passing neural network performs better than the transformer neural network, indicating the importance of edge features relative to only an adjacency matrix or distance matrix. (3) In the situation of small dataset, such as the Freesolv task with only 642 labeled molecules, CoMPT gains a 6.6% relative improvement over previous SOTAs, confirming that CoMPT model could enhance the performance on the task with few labeled data.

Task	Node Regression(MAE)	
Dataset	1H-NMR	13C-NMR
HOSE	0.33	2.85
GCN	0.28	1.43
MPNN	0.229 ± 0.002	1.355 ± 0.022
CoMPT	$\mathbf{0.214 \pm 0.003}$	$\mathbf{1.321 \pm 0.012}$

Table 2: Performance on Node-level tasks

Performance in node level task. Table 2 shows the comparison results of the baseline and CoMPT on the prediction of 1H-NMR and 13C-NMR spectra in terms of MAE. We performed experiments 5 times independently with different random seeds and report the average and standard deviation over the 5 repetitions. The average values over the 5 repetitions for 1H-NMR and 13C-NMR are 0.214 and 1.321 ppm per NMR-active atom, respectively. This indicates that our CoMPT model could extract the meaningful latent node representations and thus enable more accurate predictions of NMR spectra for new molecules.

4.3 Ablation Study (RQ2)

We conducted ablation studies on three benchmark datasets to investigate factors that influence the performance of the proposed CoMPT framework.

As shown in Table 3, CoMPT with the node position, edge direction, and message diffusion shows the best performance

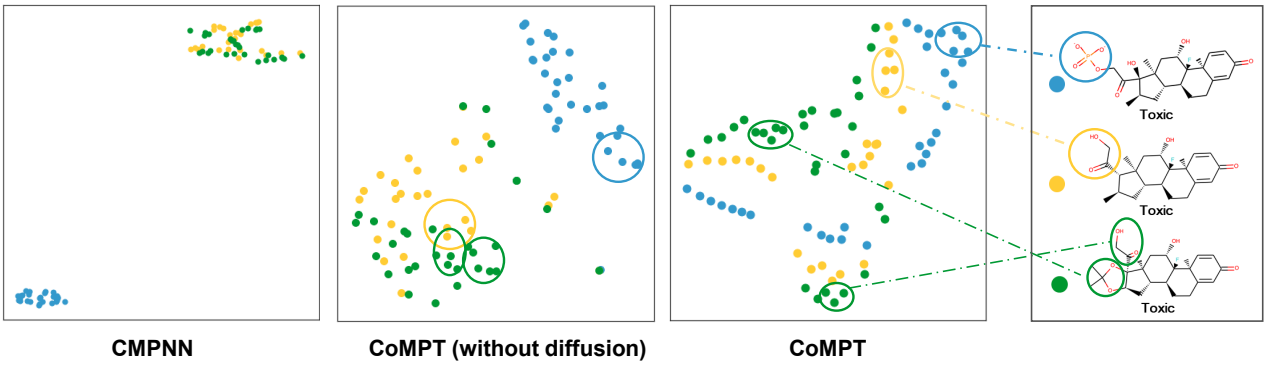

Figure 2: T-SNE visualization of atom embeddings for three similar molecules (in three colors) that have a common scaffold but various side-chains in ClinTox dataset. Ideally, the scaffold atom embeddings of these three molecules should be mixed together while the unique side-chains' embeddings should be distinguishable.

Dataset	ClinTox	Lipophilicity	1H-NMR
Without All	0.862	0.653	0.231
Without message diffusion	0.868	0.651	0.221
Without node position	0.903	0.614	0.217
Without edge direction	0.902	0.612	0.218
CoMPT	**0.934**	**0.592**	**0.214**

Table 3: Ablation results on three kinds of datasets

among all architectures. The exclusion of all three modules in the "without All" variant performed the worst. The exclusion of the message diffusion mechanism caused larger decreases in performances than the ones excluding two other modules, showing the importance of reducing the message enrichment explosion. Additionally, the uses of node position and edge direction are both helpful for the final performance.

4.4 Atomic Representation Visualization (RQ3)

As shown in [Li *et al.*, 2018], the node embeddings obtained by deep GNNs tend to be over-smoothed and become indistinguishable, while shallow GNNs cannot capture atom positions within the broader context of the molecular graph. To investigate whether the CoMPT alleviated these issues as expected, we used t-distributed stochastic neighbor embedding (t-SNE) to visualize the atom embedding distributions of three similar compounds that have a common scaffold (the four-membered ring) but different side-chains. Ideally, the scaffold atom embeddings of these three molecules should be mixed together while the unique side-chains' embeddings should be distinguishable.

Figure 2 shows the projected atomic embeddings extracted from different models using the t-SNE with default settings. Overall, three methods provide reasonable results. MPNN inherits the over-smoothness issue from the GNN, making the atom embeddings indistinguishable within the graph . In contrast, both CoMPT models (with or without diffusion) can scatter the atoms well with distinguishable node embeddings. Relative to CoMPT without diffusion, CoMPT could exactly mix the scaffold atom embeddings and differentiate the side chains. More interestingly, CoMPT can distinguish the same functional groups in different chemical environments (the Hydroxy Ketones in green and in yellow, respectively). These results suggest that CoMPT can not only alleviate the over-smoothness but also capture better representations within the broader context of the molecular graph as expected.

In the node-level tasks, our CoMPT model reaches 0.214 MAE that many molecules with densely packed 1H-NMR spectra can be resolved at these levels of accuracy. As an example, Figure S1 depicts the structure of the 3-Formylbenzoic acid, which has 6 hydrogen atoms, labeled from 11 to 16 with peaks within the 4-14 ppm range. The small difference between the ground truth and the prediction further proves that our model has a good capability in the node-level tasks.

5 Conclusions

In this paper, we propose a Communicative Message Passing Transformer (CoMPT) neural network to improve the molecular representation by reinforcing the message interactions between nodes and edges based on the Transformer model. Further, we introduce a message diffusion mechanism to decay the message enrichment explosion as well as over-smoothness during the message passing process. Extensive experiments demonstrate that our CoMPT model obtains superior performance against state-of-the-art baselines on both graph-level tasks and node-level tasks.

Acknowledgments

This work has been supported by the National Key R&D Program of China(2020YFB0204803), National Natural Science Foundation of China(61772566), Guangdong Key Field R&D Plan(2019B020228001, 2018B010109006), Introducing Innovative and Entrepreneurial Teams(2016ZT06D211), Guangzhou S&T Research Plan(202007030010).

References

[Chen *et al.*, 2019] Benson Chen, Regina Barzilay, and Tommi Jaakkola. Path-augmented graph transformer network. *arXiv preprint arXiv:1905.12712*, 2019.

[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.

[Honda *et al.*, 2019] Shion Honda, Shoi Shi, and Hiroki R Ueda. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv preprint arXiv:1911.04738*, 2019.

[Jonas and Kuhn, 2019] Eric Jonas and Stefan Kuhn. Rapid prediction of nmr spectral properties with quantified uncertainty. *Journal of cheminformatics*, 11(1):1–7, 2019.

[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.

[Kwon *et al.*, 2020] Youngchun Kwon, Dongseon Lee, Youn-Suk Choi, Myeonginn Kang, and Seokho Kang. Neural message passing for nmr chemical shift prediction. *Journal of Chemical Information and Modeling*, 60(4):2024–2030, 2020.

[Li *et al.*, 2018] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[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.

[Lu *et al.*, 2019] 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.

[Maziarka *et al.*, 2020] Łukasz Maziarka, Tomasz Danel, Sławomir Mucha, Krzysztof Rataj, Jacek Tabor, and Stanisław Jastrzębski. Molecule attention transformer. *arXiv preprint arXiv:2002.08264*, 2020.

[Ramsundar *et al.*, 2015] 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.

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

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

[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.

[Song *et al.*, 2020] Ying Song, Shuangjia Zheng, Zhangming Niu, Zhang-Hua Fu, Yutong Lu, and Yuedong Yang. Communicative representation learning on attributed molecular graphs. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,(IJCAI 2020)*, pages 2831–2838, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

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

[Xiong *et al.*, 2019] Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of Medicinal Chemistry*, 2019.

[Xiong *et al.*, 2020] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. *arXiv preprint arXiv:2002.04745*, 2020.

[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.

[Zhang and Meng, 2019] Jiawei Zhang and Lin Meng. Gresnet: Graph residual network for reviving deep gnns from suspended animation. *arXiv preprint arXiv:1909.05729*, 2019.