Electronic supplementary information for learning size-adaptive molecular substructures for explainable drug-drug interaction prediction by substructure-aware graph neural network

Ziduo Yanga#, Weihe Zhonga#, Qiujie Lv#, Calvin Yu-Chian Chenab,c*

a Artificial Intelligence Medical Center, School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, 510275, China
b Department of Medical Research, China Medical University Hospital, Taichung 40447, Taiwan
c Department of Bioinformatics and Medical Engineering, Asia University, Taichung, 41354, Taiwan

# Equal contribution

* Corresponding Authors

Calvin Yu-Chian Chen, Ph.D.

Dean of Artificial Intelligence Medical Center, Professor of School of Intelligent Systems Engineering, Sun Yat-sen University.

TEL: 02039332153

E-mail: chenyuchian@mail.sysu.edu.cn
S1. Supplemental Methods

S1.1 Problem formulation

The DDI prediction task is to develop a computational model that receives two drugs with an interaction type as inputs and generates an output prediction indicating whether there exists an interaction between them. Formally, given a set of drugs $\mathcal{D}$, a set of interaction types $\mathcal{I} = \{I_i\}_{i=1}^{K}$, and a dataset $\mathcal{M} = \{(d_x, d_y, r)\}_{i=1}^{N}$, where $d_x$ and $d_y$ are sampled from $\mathcal{D}$ with interaction type $r \in \mathcal{I}$, and $K$ is the total interaction types and $N$ is the sample numbers, our goal is to find a model $f : \mathcal{D} \times \mathcal{D} \times \mathcal{I} \rightarrow \{0,1\}$ that maps the input into the binary decision.

S1.2 Input representation

The model takes DDI tuples $(d_x, d_y, r)$ as input. A drug $d$ is represented as a molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges/bonds. In a molecule, $v_i \in \mathcal{V}$ is the $i$-th atom and $e_{ij} \in \mathcal{E}$ is the chemical bond between $i$-th and $j$-th atoms. Each node $v_i$ has a corresponding feature vector $x_i \in \mathbb{R}^{d_i}$ and each edge/bond $e_{ij}$ has a feature vector $x_{ij} \in \mathbb{R}^{d_{e}}$. The features used for atoms and bonds can be found in Tables S1 and S2 of ESI.

S1.3 Directed message passing neural network for substructures extraction

The D-MPNN$^1$ is a variant of the generic message passing neural network (MPNN)$^2$ architecture. Therefore, we first introduced the MPNN and then generalized it to D-MPNN.

S1.3.1 Message passing neural network

The MPNN is some type of GNN that maps an undirected graph $\mathcal{G}$ to a graph-
level vector $h_v$ usually with a message passing phase and a readout phase. The message passing phase updates each node by considering its neighboring nodes, and the readout phase computes a graph-level feature vector $h_g$ for the whole graph. Concretely, the message passing phase consists of $T$ steps. On each step $t$, node-level hidden features $h_i^{(t)}$ and messages $m_i^{(t)}$ associated with each node $v_i$ are updated using the message function $M_t$ and node update function $U_t$ according to

$$m_i^{(t+1)} = \sum_{v_j \in N(v_i)} M_t(h_j^{(t)}, h_j^{(t)}, e_{ij})$$

$$h_i^{(t+1)} = U_t(h_i^{(t)}, m_i^{(t+1)})$$

where $N(v_i)$ represents the set of neighbors of $v_i$ in the graph $G$, and $h_i^{(0)}$ is set to the initial atom features $x_i$. The readout phase then uses a readout function $R$ to obtain a graph-level feature vector based on the node-level features at the final step as follows

$$h_g = R(\{h_i^{(T)} | v_i \in G\})$$

The message function $M_t$, node update function $U_t$, and readout function $R$ are all learned differentiable functions.

**S1.3.2 Directed message passing neural network**

In the MPNN, every node will have received a message from all of its neighbors. However, such a mechanism is likely to introduce noise into the graph representation. Using Figure S1 (a) as an illustration, in node-level message passing, the node $v_1$ aggregates the message from its neighbors (i.e., $v_2$, $v_3$, and $v_4$); however, in the next iteration, the message of the node $v_1$ will be propagated to the node $v_4$ that already contains the information of node $v_1$ in the previous iteration, which creates an
unnecessary loop in the message passing trajectory.

The D-MPNN propagates messages along with directed bonds instead of nodes, as shown in Figure S1 (b). To highlight the difference between \( e_{ij} \) and \( e_{ji} \) (i.e., two different directions of a bond), we renamed them as \( e_{i \rightarrow j} \) and \( e_{j \rightarrow i} \). Formally, the D-MPNN operates on bond-level hidden features \( h_{ij}^{(t)} \) and messages \( m_{ij}^{(t)} \) instead of \( h_i^{(t)} \) and \( m_i^{(t)} \) used in MPNN. Note that \( h_{ij}^{(t)} \) and \( m_{ij}^{(t)} \) are distinct from \( h_i^{(t)} \) and \( m_i^{(t)} \), where the former is bond-level features along with the bond \( e_{i \rightarrow j} \) while the latter are bond-level features along with the bond \( e_{j \rightarrow i} \). The corresponding message passing update equations are thus

\[
m_{ij}^{(t+1)} = \sum_{k \in N(i) \cap N(j)} M_i(v_k, v_j, h_{ij}^{(t)})
\]

(4)

\[
h_{ij}^{(t+1)} = U_i(h_{ij}^{(t)}, m_{ij}^{(t+1)})
\]

(5)

Note that \( m_{ij}^{(t+1)} \) does not depend on its reverse message \( m_{ji}^{(t)} \) from the previous iteration, which leads to a more efficient message passing compared with MPNN\(^1\). We implemented functions \( M_i \) and \( U_i \) as follows

\[
M_i(v_i, v_j, h_{ij}^{(t)}) = h_{ij}^{(t)}
\]

(6)

\[
U_i(h_{ij}^{(t)}, m_{ij}^{(t+1)}) = h_{ij}^{(t)} + m_{ij}^{(t+1)}
\]

(7)

Note that Equations (6) and (7) do not apply any transformations to feature vectors at each iteration \( t \), which can be viewed as an instance of the concept of simplifying graph convolutions\(^3\). Besides, we proposed to initialize the bond-level hidden features as

\[
h_{ij}^{(0)} = W_i x_i + W_j x_j + W_{ij} x_{ij}
\]

(8)

where \( W_i \in \mathbb{R}^{h \times d} \), \( W_j \in \mathbb{R}^{h \times d} \), and \( W_{ij} \in \mathbb{R}^{h \times d'} \) are learnable weight matrices. After \( T \)
times updates of bond-level hidden features, we returned to the node-level hidden features of the molecule by summing the incoming bond-level features according to the following:

\[ h_i^{(T)} = \sum_{j \in N(i)} h_{ij}^{(T)} \]  \hspace{1cm} (9)

The readout phase of the D-MPNN is the same as the readout phase of a generic MPNN as described in Equation (3). The original D-MPNN used a global sum pooling function to obtain the graph-level representation \( h_G \) for a given molecule/graph \( G \).

**Reference**


## S2. Supplemental Tables

### Table S1. Atom features.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atom type</td>
<td>Heavy atom type (e.g., C, O, N, S, I)</td>
<td>Total number of heavy atoms in the dataset*</td>
</tr>
<tr>
<td>Degree</td>
<td>Number of covalent bonds [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]</td>
<td>11*</td>
</tr>
<tr>
<td>Implicit valence</td>
<td>Implicit valence of the atom [0, 1, 2, 3, 4, 5, 6, 7]</td>
<td>7*</td>
</tr>
<tr>
<td>Hybridization</td>
<td>[sp, sp2, sp3, sp3d, sp3d2]</td>
<td>5*</td>
</tr>
<tr>
<td>Aromatic</td>
<td>Whether the atom is part of an aromatic system</td>
<td>1</td>
</tr>
<tr>
<td>Formal charge</td>
<td>Formal charge of the atom</td>
<td>1</td>
</tr>
<tr>
<td>Radical electrons</td>
<td>Number of radical electrons for the atom</td>
<td>1</td>
</tr>
</tbody>
</table>

*One-hot representation

### Table S2. Bond features.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bone type</td>
<td>[single, double, triple, aromatic]</td>
<td>4*</td>
</tr>
<tr>
<td>Conjugated</td>
<td>Whether the bond is part of a conjugated system</td>
<td>1</td>
</tr>
<tr>
<td>Ring</td>
<td>Whether the bond is part of a ring</td>
<td>1</td>
</tr>
</tbody>
</table>

*One-hot representation

### Table S3. Search range and selected values of hyperparameters for SA-DDI.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Search range</th>
<th>Selected value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of iterations $T$</td>
<td>[1, 2, 3, 4, 5, 6, 10, 15, 20, 25]</td>
<td>10</td>
</tr>
<tr>
<td>Number of hidden units for $h_i/h_j$</td>
<td>[32, 64, 128]</td>
<td>64</td>
</tr>
<tr>
<td>Learning rate</td>
<td>[1e-4, 5e-4, 1e-3, 1e-2]</td>
<td>1e-3</td>
</tr>
</tbody>
</table>
Figure S1. The difference between MPNN and D-MPNN. The messages are propagated through (a) nodes for MPNN and (b) bonds for D-MPNN.
Figure S2. Distribution of DDI types of the (a) DrugBank dataset and (b) TWOSIDES dataset.
Figure S3. Visualization of the key substructures for DDIs (a) between dicoumarol and milnacipran, thiamylal, and (b) between apo-carbamazepine and cimetidine, verapamil.
Figure S4. Visualization of the key substructures for DDIs between dicoumarol and the other seven drugs using the cold start setting which removes dicoumarol from the training set. The center of the most important substructure and its receptive field are shown as blue and orange colors respectively.