**A Self-Attention Based Message Passing Neural Network for Predicting Molecular Lipophilicity and Aqueous Solubility**

*Bowen Tang1,2, Skyler T. Kramer 2, Meijuan Fang1, Yingkun Qiu1, Zhen Wu1,\* and Dong Xu2,\**

1Fujian Provincial Key Laboratory of Innovative Drug Target Research, School of Pharmaceutical Sciences, Xiamen University, Xiamen 361000, China

2 Department of Electrical Engineering and Computer Science, Informatics Institute, and Christopher S. Bond Life Sciences Center, University of Missouri, Columbia, MO 65211, USA

\*Correspondence: wuzhen@xmu.edu.cn，xudong@missouri.edu

**Supporting Information**

|  |
| --- |
| Algorithm 1: Edge-dependent message passing neural network with self-attention |

 **Table S1.** Algorithm of SANMP model
1: **Input:** clean SMILES in batch

2: **Create molecule graph:** for each molecule mol: list(node\_index), list(edge\_index), list(n2e),list(e2n) #index mapping n2e: node to edge, e2n: edge to node3: **Initialize node features:** for each node *a*: F*a* = *f(a)*

4: **Initialize edge features:** for each edge *b*: F*b* = *f(b)*

5: **Initialize message:** message=Re(Winp ∙ concatenate(F*a*, F*b*)) # number equals the edges

6: # Message passing

7: **for** 2~ step - 1:

8: node\_neighbors\_message = select(message, n2e) #none message-receiving neighbor

9: neighbors\_sum= (Node\_neighbors\_message)

10: message= Re(Winp ∙ concatenate(F*a*, F*b*) + Wh ∙ neighbors\_sum)

11: #Messsage readout

12: node\_neighbors\_message = select(message, n2e) #all the neighbors

13: hidden state of nodes = Re(concatenate(F*a*, node\_neighbors\_message))

14: Wscore, *EG* = Self-attention(*G*) 15: Mol\_vector= Re (*Wo* ∙ global\_pool(G+*EG*))16: #Prediction17: F*out* = DenseNetworks(Mol\_vector )

|  |
| --- |
| Note: “step” denotes the number of message passing steps; *Wh* denotes the hiddenweights; Redenotes the ReLU activate function. |

****

B

A

**Fig S1**. Data distributions of lipophilicity and solubility.



**Fig S2**. Molecular heat map for optimizing the lipophilicity.

**Note S1.** Equations of metrics used to compare performances.

$$MSE={(\sum\_{i=1}^{n}\left(Y\_{i}-\hat{Y\_{i}}\right)^{2})}/{n} \left(1\right)$$

$$RMSE=\sqrt{MSE} \left(2\right)$$

$$MAE={(\sum\_{i=1}^{n}\left|Y\_{i}-\hat{Y\_{i}}\right|)}/{n} \left(3\right)$$

$$R^{2}=1-{\left(MSE\*n\right)}/{\left(\sum\_{i=1}^{n}\left(Y\_{i}-\overbar{Y}\right)^{2}\right)}, performed on training set \left(4\right)$$

$$Q^{2}=R^{2}, performed on testing set \left(5\right)$$

$$PC=\frac{n\left(\sum\_{}^{}Y\_{i}\hat{Y\_{i}}\right)-(\sum\_{}^{}Y\_{i})(\sum\_{}^{}\hat{Y\_{i}})}{\sqrt{[n\sum\_{}^{}Y\_{i}^{2}-\left(\sum\_{}^{}Y\_{i}\right)^{2}][n\sum\_{}^{}\hat{Y\_{i}}^{2}-\left(\sum\_{}^{}\hat{Y\_{i}}\right)^{2}]}} \left(6\right)$$

Note that *Yi* refers to the ground truth label, where the hatted variant is the predicted label. Also note that (1), (2), (3), and (6) can be calculated for the training and testing sets and retain the original name of the metric. Conversely, the metrics in (4) and (5) have different names when computed on the training/testing set, respectively.